ASPER – ASSOCIAÇÃO PARAIBANA DE ENSINO RENOVADO

COORDENAÇÃO DE CIÊNCIAS DA COMPUTAÇÃO

VALBERTO VIEIRA CARNEIRO

VANJHONY COSMO DA SILVA AMÂNCIO

YGOR MIKASSIO DUARTE LINS

PROCESSO DE DESENVOLVIMENTO DE SOFTWARE:

UM ESTUDO PRÁTICO PARA MICRO E PEQUENAS EMPRESAS DE TI

JOÃO PESSOA

2009

VALBERTO VIEIRA CARNEIRO

VANJHONY COSMO DA SILVA AMÂNCIO

YGOR MIKASSIO DUARTE LINS

PROCESSO DE DESENVOLVIMENTO DE SOFTWARE:

UM ESTUDO PRÁTICO PARA MICRO E PEQUENAS EMPRESAS DE TI

Trabalho de conclusão de curso apresentado à Associação Paraibana de Ensino Renovado como requisito parcial para a obtenção do título de Bacharel em Ciências da Computação.

Orientador: Maxwell Anderson I. Amaral

JOÃO PESSOA

2009

Ficha catalográfica

VALBERTO VIEIRA CARNEIRO

VANJHONY COSMO DA SILVA AMÂNCIO

YGOR MIKASSIO DUARTE LINS

PROCESSO DE DESENVOLVIMENTO DE SOFTWARE:

UM ESTUDO PRÁTICO PARA MICRO E PEQUENAS EMPRESAS DE TI

Aprovado em: _____ de _____ de _____

BANCA EXAMINADORA

_____

Componente da banca – Instituição

_____

Componente da banca – Instituição

_____

Componente da banca – Instituição

Valberto:

Dedico este trabalho à minha mãe Eliete que me acompanha durante toda a minha vida, com amor e perseverança.

Também dedico à minha professora e amiga Josemary pela contribuição profissional que sempre me ajudou a crescer em todas as esferas da vida.

Ygor:

Agradeço aos meus pais que sempre me incentivaram e me ajudaram a seguir em frente em busca da realização dos meus desejos.

A minha namorada pela sua paciência e apoio, mas principalmente por suas opiniões nas tomadas de decisões importantes.

A professora Josemary Freire pela atenção, dedicação e apoio ao longo deste e de outros trabalhos realizados ao longo do curso.

Vanjhony:

Esse é o termino de uma fase da minha vida, eu agradeço a todas as pessoas que fizeram parte dela e a tornaram possível, meus pais, meu irmão, minha menina, meus amigos, meus mestres, meus companheiros. Todos nós, formandos do curso de Ciência da Computação, iniciado no ano 2006.1, agora, fazemos parte um da vida do outro, uns com mais intensidade que outros, mas, com certeza, todos tem uma parcela de importância nessa conquista que obtivemos hoje.

AGRADECIMENTOS

- Agradecemos a Deus pela vida. Em poder almejar novos objetivos de crescimento através do conhecimento adquirido.

- Aos nossos pais, através do amor incondicional, da educação e formação cultural, pelas quais obtivemos nossa conduta moral.

- As nossas famílias que colaboraram nesta jornada, tanto em âmbito psicológico como financeiro.

- Agradecemos aos companheiros de equipe, agora amigos, pelo trabalho árduo durante os vários fins de semana seguidos dedicados ao desenvolvimento do conhecimento contido neste trabalho.

- A todos os professores que contribuíram para a nossa formação profissional.

- Em especial à Josemary e Jader pelo apoio constante nessa jornada, contribuindo para o fortalecimento dos conceitos profissionais e pessoais.

- Ao nosso professor e orientador Maxwell Anderson, que foi de fundamental importância nesse trabalho, sempre nos estimulando e mostrando o verdadeiro potencial da Ciência da Computação e da Engenharia de Software.

- Aos “celebrados” pelo constante estímulo em busca da dominação global, sempre adicionando ideias peculiares que nos motivaram a evoluir constantemente.

- Enfim, agradecemos a todos os amigos que, direta ou indiretamente, contribuíram para mais esta conquista. Muitíssimo obrigado a todos.

LISTA DE FIGURAS

Figura 3.1 - Ciclo de vida do modelo sequencial (cascata) 19
Figura 3.2 - Comparação entre os modelos em cascata, incremental e iterativo 20
Figura 3.3 - Modelo de processo em espiral (PRESSMAN, 2006, p. 45). 22
Figura 3.4 - Arquitetura organizacional do RUP 24
Figura 4.1 - Visualização do Prumo em níveis 28
Figura 4.2 - Ciclo de vida do Prumo 30
Figura 4.3 - Exemplo de representação de uma atividade 45

LISTA DE GRÁFICOS

Gráfico 1.1 - Previsão de crescimento mundial do setor de software para 2005/2009 (Fonte: Abes/IDC) 11
Gráfico 2.1 – Distribuição do mercado brasileiro de software de acordo com a classificação das empresas.....14

LISTA DE TABELAS

Tabela 2.1 - Classificação das micro e pequenas empresas de software brasileiras segundo o SEBRAE 15

LISTA DE SIGLAS E ABREVIATURAS

ABES – Associação Brasileira das Empresas de Software

ASSESPRO - Associação das Empresas Brasileiras de Tecnologia da Informação, Software e Internet

BID- Banco Interamericano de Desenvolvimento

CMMI - *Capability Maturity Model Integration*

DER – Diagrama Entidade-relacionamento

ER – Entidade Relacional (diagrama)

FINEP - Financiadora de Estudos e Projetos

GMS – Gerenciamento de Mudança de Software

GQS – Gerenciamento de Qualidade de Software

ISO – *International Organization for Standardization*

IBM - *International Business Machines*

MA-MPS – Método de Avaliação para Melhoria do Processo de Software

MCT – Ministério da Ciência e Tecnologia

MPS.BR – Melhoria do Processo de Software Brasileiro

POO – Programação Orientada à Objetos (paradigma)

RUP – *Rational Unified Process*

SEBRAE - Serviço Brasileiro de Apoio às Micro e Pequenas Empresas

SGC – Software para gerenciamento de crediários

SOFTEX – Sociedade Brasileira para a Promoção da Exportação de Software

RESUMO

Este trabalho apresenta uma proposta de um processo de desenvolvimento de software para micro e pequenas empresa de tecnologia da informação. Sua construção foi motivada a partir da necessidade de um grupo de universitários, com um cliente real, que necessitava de um software para ajudar a gerenciar da sua empresa. Em busca de um desenvolvimento concreto, sistemático, que garanta de uma boa qualidade do produto de software, eles buscaram nos conhecimentos científicos adquiridos, respostas e soluções para alcançar tais objetivos. Foi então que nasceu o Prumo, para guiá-los rumo a uma solução fácil, eficiente e mercadologicamente competitiva.

A partir dele, também foram realizadas pesquisas de campo e bibliográficas, observando características do mercado-alvo. Veio a surpresa! Não só eles, mas uma grande fatia do mercado brasileiro de software é composta por micro e pequenas empresas que precisam adicionar qualidade ao seu processo de produção de software. Por isso, esta proposta se torna também um estudo que pode ser amplamente utilizado para reduzir custos e melhorar a distribuição dos recursos de uma micro e pequena empresa de TI. Com o objetivo de embasar teoricamente os leitores foram discutidos conceitos de alguns dos principais autores (Pressman, Sommerville), processos, metodologias (sequencial, interativo e incremental, espiral, ágil) e *frameworks* (RUP) sobre o assunto na atualidade, de forma que o leitor terá uma maior intimidade com os termos e técnicas que serão usados no decorrer do texto.

Ao final o leitor terá adquirido uma carga de conhecimentos que lhe proporcionará entender os objetivos e os métodos utilizados no processo de forma integral. Além disso, terá em mãos um processo de desenvolvimento de software que mostrará formas de como aproveitar melhor seus recursos, proporcionando um aumento na qualidade dos serviços prestados e na produtividade da empresa como um todo.

ABSTRACT

This paper presents a proposal for a process of software development for micro and small enterprise information technology. Its construction was motivated from the need for a group of students, with a real customer, who needed a software to help manage your company. In search of a concrete development, systematic, ensuring a good quality of the software product, they sought the knowledge gained, answers and solutions to achieve these goals. It was then that the Plummet to guide them towards an easy, efficient and competitive merchandising.

From it, were also carried out field research and literature, noting characteristics of the target market. Then surprise! Not only them, but the vast majority of the Brazilian software market consists of micro and small businesses that need to add quality to the process of software production. Therefore, this proposal becomes also a study that can be widely used to reduce costs and improve resource distribution of micro and small business IT. With the aim of theoretical readers were discussed concepts of some major authors (Pressman, Sommerville), processes, methods (sequential, interactive, incremental, spiral, agile) and frameworks (RUP) on the subject today, so that the reader will have a greater intimacy with the terms and techniques that will be used throughout the text.

At the end the reader will have acquired a load of knowledge that will provide you understand the goals and methods used in the process comprehensively. They will also have a hand in the process of developing software that will show ways of how to best leverage their resources by providing an increase in service quality and productivity of the company as a whole.

SUMÁRIO

1 INTRODUÇÃO 5
1.1 SITUAÇÃO ENCONTRADA 7
1.2 JUSTIFICATIVA 8
1.3 OBJETIVOS GERAIS 8
1.4 OBJETIVOS ESPECÍFICOS 9
1.5 FUNDAMENTAÇÃO TEÓRICA 9
2 MICRO E PEQUENAS EMPRESAS DE SOFTWARE 13
2.1 MERCADO 13
2.2 PERFIL DAS EMPRESAS 14
2.3 ORGANIZAÇÃO 14
2.3.1 Colaboradores 15
2.4 PROBLEMAS DA EMPRESA DE DESENVOLVIMENTO 15
3 PROCESSO DE DESENVOLVIMENTO DE SOFTWARE 17
3.1 DEFINIÇÃO DE PROCESSO 17
3.2 DEFINIÇÃO DE PROCESSO DE SOFTWARE 17
3.3 MODELOS DE PROCESSOS 18
3.3.1 Modelo em Cascata 18
3.3.2 Modelo iterativo e incremental 19
3.3.3 Diferenciando os modelos em cascata, iterativo e incremental 20
3.3.4 Modelo em espiral 21
3.3.5 Modelo Ágil 22
3.4 PADRÕES POR EXCELÊNCIA 23
3.4.1 *Rational Unified Process* (RUP) 24
3.4.2 MPS.Br 25
4 O PRUMO 27
4.1 NÍVEIS DE DETALHAMENTO DO PROCESSO 27
4.2 NÍVEL DO CICLO DE VIDA 29
4.2.1 Fase de solicitação 31
4.2.2 Fase de análise 31
4.2.3 Fase de planejamento 31
4.2.4 Fase de Desenvolvimento 32

4.2.5 Fase de Teste 33
4.2.6 Fase de Entrega 33
4.2.7 Verificação da qualidade 34
4.2.8 Gerenciamento de mudança 34
4.3 OS PAPÉIS, RESPONSABILIDADES E ARTEFATOS 35
4.3.1 Cliente 36
4.3.2 Analista de Requisitos 37
4.3.3 Gerente de Projetos 39
4.3.4 Arquiteto de Software 39
4.3.5 Desenvolvedor 40
4.3.6 Analista de Testes 41
4.3.7 Analista de Suporte 41
4.3.8 Analista de Qualidade 42
4.4 CONFROTAMENTO DE PAPÉIS 43
4.4.1 Distribuindo papéis em uma microempresa 43
4.5 NÍVEL DAS ATIVIDADES 44
4.5.1 Atividades da fase de solicitação 45
4.5.1.1 Solicitar serviço 45
4.5.1.2 Analisar necessidade 46
4.5.1.3 Negociar proposta de solução 46
4.5.2 Atividades da fase de análise 47
4.5.2.1 Elicitar requisitos 47
4.5.2.2 Revisar requisitos 48
4.5.2.3 Definir tecnologia 48
4.5.2.4 Avaliar riscos e mensurar requisitos 48
4.5.2.5 Definir diretrizes de teste 49
4.5.2.6 Negociar requisitos 49
4.5.3 Atividades da fase de planejamento 50
4.5.3.1 Planejar desenvolvimento 50
4.5.3.2 Preparar ambiente de desenvolvimento 50
4.5.4 Atividades da fase de desenvolvimento 51
4.5.4.1 Definir arquitetura 51
4.5.4.2 Codificar 51
4.5.5 Atividades da fase de teste 51
4.5.5.1 Testar 51

4.5.6 Atividades da fase de entrega 53
4.5.6.1 Planejar treinamento e integração 53
4.5.6.2 Treinar usuários 53
4.5.6.3 Implantar solução e dar suporte 54
4.6 PROCESSOS DE APOIO 54
4.6.1 Gerenciamento das Mudanças (GMD) 54
4.6.1.1 Analisar solicitação 55
4.6.1.2 Negociar solução com o cliente 55
4.6.1.3 Executar mudança planejada 55
4.6.2 Verificação da qualidade (VQA) 56
4.6.2.1 Levantamento do projeto 56
4.6.2.2 Avaliação da qualidade 56
4.6.2.3 Reunião 57
4.6.2.4 Correção 57
5 CONSIDERAÇÕES FINAIS 58
REFERÊNCIAS 59
APÊNDICE A - CICLO DE VIDA E MAPA DE ATIVIDADES 63
APÊNDICE B – PAPÉIS 69
APÊNDICE B.1 – CLIENTE 71
APÊNDICE B.2 – ANALISTA DE REQUISITOS 75
APÊNDICE B.3 – GERENTE DE PROJETOS 79
APÊNDICE B.4 – ARQUITETO DE SOFTWARE 83
APÊNDICE B.5 – DESENVOLVEDOR 87
APÊNDICE B.6 – ANALISTA DE TESTE 91
APÊNDICE B.7 – ANALISTA DE SUPORTE 95
APÊNDICE B.8 – ANALISTA DE QUALIDADE 99
APÊNDICE C – ARTEFATOS 103
APÊNDICE C.1 – SOLICITAÇÃO 105
APÊNDICE C.2 – DOCUMENTO DE VISÃO 109
APÊNDICE C.3 – GLOSSÁRIO 135
APÊNDICE C.4 – REGRAS DE NEGÓCIO 147
APÊNDICE C.5 – ATA DE REUNIÃO 159
APÊNDICE C.6 – DOCUMENTO DE ESPECIFICAÇÃO DE REQUISITOS 163
APÊNDICE C.7 – DOCUMENTO DE ESPECIFICAÇÃO DE CASO DE USO 179
APÊNDICE C.8 – PROTÓTIPO DA INTERFACE COM O USUÁRIO 195

APÊNDICE C.9 – MATRIZ DE RASTREABILIDADE.....205
APÊNDICE C.10 – LISTA DE RISCOS 209
APÊNDICE C.11 – PLANO DE PROJETO.....213
APÊNDICE C.12 – DOCUMENTO DE ARQUITETURA DE SOFTWARE.....229
APÊNDICE C.13 – MODELO DE IMPLEMENTAÇÃO 249
APÊNDICE C.14 – PLANO DE INTEGRAÇÃO DO BUILD.....261
APÊNDICE C.15 – PLANO DE TESTE 273
APÊNDICE C.16 – PLANO DE IMPLANTAÇÃO 299
APÊNDICE C.17 – PLANO DE TREINAMENTO.....311
APÊNDICE C.18 – GESTÃO DE AÇÕES CORRETIVAS 327
APÊNDICE C.19 – LISTA DE VERIFICAÇÃO DA QUALIDA.....331

1 INTRODUÇÃO

Como parte dos acontecimentos naturais sobre a Terra, sua evolução tem sido observada por vários estudiosos e pesquisadores no mundo inteiro. Na “era da informação” esta evolução não poderia ser diferente. Durante a década de 80, quando o desenvolvimento de novas arquiteturas de computadores priorizava o aumento da capacidade de processamento e os custos de produção eram o foco dos pesquisadores, surgiam vários rumores sobre a crise de software. Jornais em várias partes do mundo anunciavam a decadência dos softwares produzidos. Nesse contexto, as pessoas começavam a observar que profundas mudanças na sociedade estariam por vir. Naisbitt previu a transformação da sociedade industrial em uma sociedade da informação, gerando profundos impactos sobre as pessoas no mundo contemporâneo (PRESSMAN, 1995, p. 4).

Passados alguns anos de crise, o software começou a reagir a partir de um quadro crítico, evoluindo cada vez mais através de novas técnicas de desenvolvimento que surgiam a cada ano, em meados da década de 90. A adoção às novas práticas de desenvolvimento de software era perceptível e com a popularização da Internet, a distribuição desses conhecimentos ganhou o mundo rapidamente. Assim várias nações começavam a identificar-se na produção de tipos específicos de software. Os Estados Unidos na produção de software *standard*[1]; a Europa se tornava líder dos softwares customizados[2] e o Japão, na produção de soluções para o setor financeiro e jogos eletrônicos.

Entre essas e outras razões é que as pessoas começaram a “dar o pontapé inicial” na criação de uma nova empresa para desenvolvimento de soluções de software. Então, grupos de estudantes, curiosos e pessoas com experiência adquiridas em empresas privadas começaram a organizar seus conhecimentos para produzir tecnologia e novas soluções para o mundo.

---

[1] Engloba aqueles em que há possibilidade de instalação pelo próprio usuário (FILHO, HOCHSTETLER, *et al.*, 2006, p. 5)

[2] Nesse caso, pode ter a mesma definição de software sob demanda, aquele que é desenvolvido de acordo com as especificações de um determinado cliente (FILHO, HOCHSTETLER, *et al.*, 2006, p. 5)

Nesse ritmo, pequenas empresas de desenvolvimento abriam suas portas a todo o momento e junto a esse crescimento também surgiam novos problemas na organização, padronização das atividades e na construção dos produtos. Em contra partida os problemas dos clientes ficavam cada vez mais complexos, aumentando mais ainda a dificuldade na hora de desenvolver ou integrar sistemas.

Para tentar ajudar às micro e pequenas empresas de software brasileiras, um modelo de implementação de processo para desenvolvimento de software foi desenvolvido, observando-se o perfil e os recursos dessas empresas. A partir de um estudo detalhado, nasceu o Prumo com uma proposta de mostrar que é possível desenvolver software com qualidade, sem que sejam necessárias grandes equipes compostas de pessoas altamente qualificadas para isso.

O Prumo possibilita que micro empresas, com menos de dois anos de atividade e com uma disponibilidade de recursos limitado, possam utilizá-lo de forma prática e eficiente. Para isso, ele utiliza um método de controle eficiente, sem sobrecarregar a equipe e mantendo apenas os artefatos[3] mais importantes para o desenvolvimento de software com qualidade.

Além disso, outra de suas características importantes é a distribuição dos papéis disponíveis para uma equipe com um número menor de pessoas, enfatizando a sua capacidade de adaptação às diferentes equipes, encontradas facilmente no mercado. A importância dessa proposta deve-se às necessidades intrínsecas do mercado capitalista em busca de profissionais com capacidade para atuar em diversas atividades de acordo com a demanda exigida.

Portanto, o Prumo surgiu como uma alternativa para aumentar as expectativas desse mercado em desenvolver software com qualidade, atendendo às exigências do mercado nacional e internacional, podendo ser utilizado sem que seja necessário investimentos altos em consultoria.

[3] Artefatos são todas as especificações desenvolvidas durante um processo de software, essa definição envolvem documentos, diagramas, protótipos ou qualquer outra informação necessário ao processo (MELO, 2004, p. 5) (MELO, 2004, p. 5).

1.1 SITUAÇÃO ENCONTRADA

Uma característica importante observada no mercado de criação de software é a não padronização dos processos de desenvolvimento, onde, a maioria das empresas possui pouca ou nenhuma especificação formal para as suas atividades. Isso pode ocorrer devido à escassez de recursos financeiros disponíveis para investimentos na organização dos processos ou mesmo a falta de conhecimento. Isso deixa os produtos e serviços oferecidos por elas ficarem abaixo da qualidade exigida pelo mercado nacional e internacional.

Outra característica observada nas empresas que não possuem seus processos definidos é a personalização das atividades, por exemplo: o profissional que desempenha a tarefa de codificação do software pode criar seus próprios padrões de codificação, dando nomes para variáveis, procedimentos e funções da forma como ele considerar mais conveniente. Esse tipo de atitude não é interessante pelo fato de que a simples troca desse profissional mudará o padrão de codificação dos softwares produzidos, tornando onerosa a manutenção deles.

A característica de personalização das atividades também resulta em outro fator determinante para as empresas. Esse fator seria o tempo e o esforço para adequação e aprendizado de um novo integrante da equipe ser bem maior. Além disso, sabemos que a resistência às mudanças é uma característica intrínseca do ser humano.

Para enfatizar o exemplo, FILHO (2006, p. 9) mostra que “o avanço tecnológico no setor de informática tende a ser maior que o progresso técnico no uso da informática”, afirmando que:

> a principal explicação para o descasamento entre o ritmo de inovação tecnológica no setor de informática e o progresso técnico no uso da informática decorre primordialmente dos custos de adaptação e aprendizado defrontados pelas empresas (FILHO, HOCHSTETLER, *et al.*, 2006, p. 9).

Portanto, se a empresa possuir um processo de desenvolvimento que segue os conceitos sugeridos pela literatura ou os modelos de processos mais usados no mercado, ela terá um ganho significativo no desempenho e na qualidade dos seus

produtos e serviços, além do risco de atraso nos prazos também diminuírem e poderem vislumbrar novos mercados, podendo alcançar até a exportação.

1.2 JUSTIFICATIVA

Tendo em vista este problema, torna-se clara a necessidade enfrentada pelas micro e pequenas empresas brasileiras de tecnologia da informação, que não possuem, ainda, um processo definido para o desenvolvimento dos seus produtos e serviços. Na maioria dos casos, boas oportunidades aparecem, mas vão embora quando é exigido mais qualidade do produto de software.

A proposta apresentada aqui se originou a partir da necessidade observada por um grupo de universitários, graduandos em Bacharelado em Ciências da Computação, que pretendem abrir uma empresa de software. O grupo trabalhava no desenvolvimento de um software comercial para empresas no interior da Paraíba que necessitava de uma boa organização para a execução de suas atividades de desenvolvimento. A partir disto, foi desenvolvido o Prumo, organizado para atender as necessidades da equipe, observando-se também as necessidades do cliente.

1.3 OBJETIVOS GERAIS

O objetivo principal é produzir um processo de software adaptado para micro e pequenas empresas brasileiras de tecnologia, que necessitem de uma sistematização de suas atividades, objetivando uma melhoria, principalmente, na qualidade dos produtos e serviços. Desta forma, os objetivos gerais são:

- Atender às demandas de mercado de software em busca de processos mais adequados à realidade das micro e pequenas empresas;
- Fazer uma junção das melhores práticas dos processos do RUP e MPS.BR;
- Esclarecer os conceitos sobre aplicações que envolvem a definição e implantação de um processo de software.

1.4 OBJETIVOS ESPECÍFICOS

Especificamente, este trabalho pretende atender as necessidades do grupo universitário citado anteriormente, na produção de um software, denominado, Software para Gerenciamento de Crediários. Devido às características da equipe serem muito semelhantes com as de uma micro empresa, pretende-se:

- Definir um processo de desenvolvimento de software voltado para micro e pequenas empresas;
- Propor uma abordagem ou modelo de processo que ajude a visualizar melhor os vários níveis de detalhamento de um processo de software;
- Definir cada nível de detalhamento do processo, de forma a esclarecer precisamente a aplicação de cada uma;
- Utilizar um estudo de caso, como justificativa para o desenvolvimento deste trabalho.

1.5 FUNDAMENTAÇÃO TEÓRICA

Sobre este assunto já foram realizadas pesquisas, abordando alguns aspectos, por outros pesquisadores, estudantes e profissionais da área de tecnologia da informação e telecomunicações. As linhas seguintes mostrarão alguns desses trabalhos e quais características se assemelham a solução apresentada, enfatizando o diferencial deste.

Um estudo encomendado pela Associação Brasileira das Empresas de Software (Abes) analisou vários aspectos do mercado de brasileiro software, sendo que, em uma das seções FILHO (2006, p. 8) observa-o como um “mercado de duas pontas”, enfatizando a importância da visão de uma empresa diante do mercado consumidor e da sua equipe de desenvolvimento. Ele menciona que os fabricantes de sistemas operacionais devem observar os dois lados do mercado para que seu produto seja aceito e utilizado amplamente. Numa ponta estão os fabricantes de aplicativos que precisam de recursos atrativos e incentivos para produzir softwares cada vez melhores. Na outra ponta estão os usuários, na expectativa de usufruir de um sistema operacional fácil e com bons aplicativos disponíveis.

O exemplo citado acima serve para reforçar a posição que as empresas de software devem tomar para que atendam as expectativas da sua equipe, oferecendo a eles uma estrutura organizacional e um processo de desenvolvimento bem definido. Consequentemente a empresa passará a produzir softwares com mais qualidade e assim atraindo cada vez mais clientes.

Outra pesquisa interessante foi desenvolvida pela Associação das Empresas Brasileiras de Tecnologia da Informação, Software e Internet (ASSESPRO), sobre políticas tributárias para o desenvolvimento do setor de tecnologia da informação. Em um dos tópicos UENO (2007, p. 5) mostra que o Brasil possui o 7º lugar mercado mundial de software com crescimento anual variando em torno dos 11% nos últimos quinze anos. Isso significa que o mercado brasileiro tem capacidade para produzir soluções interessantes não somente para o mercado local como também para o restante do mundo. Ainda na mesma pesquisa mostrou que em 2005 existiam em torno de 23 mil empresas de TI, sendo 96% delas, micro e pequenas, empregando mais de 700 mil pessoas e gerando uma receita líquida em torno de 25 milhões de reais. Isso assegura a amplitude das possibilidades de aplicação do Prumo.

FILHO (2006, p. 16) mostra a evolução no setor de software através de um panorama mundial, apontando qual o nível de significância do mercado brasileiro perante o resto do mundo. Ele apresentou a perspectiva de crescimento do setor software através do Gráfico 1.1, Nele o leitor pode ver que o Brasil ocupa a quarta melhor posição na expectativa de crescimento, em comparação com outros países bem mais desenvolvidos. Além disso, enfatiza a importância do setor de software para a economia, afirmando que entre todos os benefícios trazidos por esse avanço, o mais importante entre eles é o ganho de produtividade. Isso vale para todos os setores da indústria, comércio e serviço que focam na tecnologia como ferramenta importante para o crescimento.

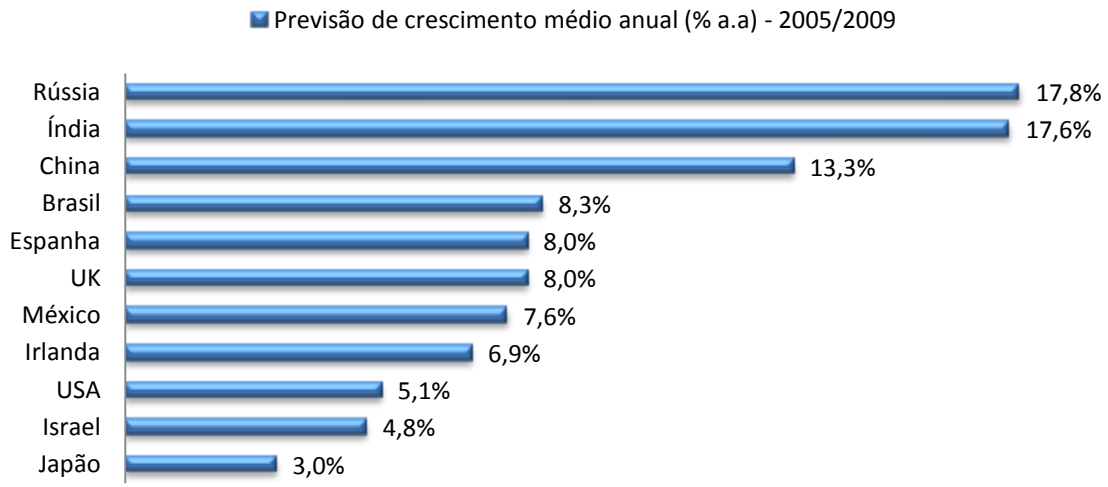

Gráfico 1.1 - Previsão de crescimento mundial do setor de software para 2005/2009 (Fonte: Abes/IDC)

Sobre os processos de desenvolvimento de software, existem inúmeras tentativas de definição, onde cada autor enfatiza uma característica diferenciada, porém todos eles caminham em direção a um ponto comum. Logo, alguns deles foram selecionados e serão mencionados procurando atingir os aspectos mais relevantes ao tema abordado neste trabalho.

Inicialmente, PRESSMAN (2006, p. 16) expõe seus conhecimentos sobre a Engenharia de Software organizando-os em camadas, onde todas são estruturadas com foco na qualidade do produto de software. Nesse modelo o alicerce é a camada correspondente ao *processo* de software, funcionando como um "adesivo que mantém unidas as camadas de tecnologia e permite o desenvolvimento racional e oportuno de softwares de computador". As outras duas camadas são: *métodos,* que compreendem as técnicas de "como fazer" para construir software e; as *ferramentas* que oferecem apoio automatizado ou semi-automatizado para o *processo* e para os *métodos*.

Adicionalmente, SOMMERVILLE (2001, p. 43), afirma que a melhoria e padronização no processo de software pode ser implementada em várias etapas, trazendo coesão aos objetivos das atividades de uma organização. Essa prática traz melhoria na comunicação entre os integrantes da equipe, redução no tempo de treinamento de um novo integrante da equipe e aumenta a economia na automação do processo.

Segundo MACHADO e WEBER (2001) a globalização da economia vem influenciando as empresas produtoras e prestadoras de serviços de software a alcançar o patamar de qualidade e produtividade internacional para enfrentarem a competitividade cada vez maior. A norma internacional NBR ISO/IEC 12207 – Tecnologia da Informação – Processos de Ciclo de Vida de Software (ISO/IEC 12207, 2008) é usada como referência em muitos países, inclusive no Brasil, para alcançar esse diferencial competitivo

2 MICRO E PEQUENAS EMPRESAS DE SOFTWARE

Esta seção apresenta um panorama do mercado brasileiro de empresas de tecnologia da informação. Mostra vários números interessantes sobre mortalidade, renda líquida e o espaço que elas ocupam e a organização. Outro ponto tratado é em relação aos recursos humanos, sistematização das atividades e quais os principais problemas.

2.1 MERCADO

Como visto anteriormente o mercado brasileiro de software é muito vasto e está em constante desenvolvimento. Segundo reportagem exibida no site do SEBRAE (BONNER, 2009), foi o setor brasileiro que mais cresceu durante a última crise que gerou instabilidade e dor de cabeça para muitos países. Apesar de o Brasil ser um país com elevada carga tributária, o governo tem lançado incentivos fiscais como redução de IPI sobre *hardware*, regulado pela Lei 8.248/91 e alterações posteriores, principalmente as leis 10176/2001 e 11.077/04. UENO (2007, p. 10) mostra que, em relação ao software, as propostas governamentais de redução de impostos ainda precisam ser bem melhoradas para se tornarem atrativas para o setor

Em se tratando de processo, a Sociedade Brasileira para a Promoção da Exportação de Software (SOFTEX), apoiada por várias instituições, entre elas o Ministério da Ciência e da Tecnologia (MCT), criaram em 2003 um modelo de referência para Melhoria do Processo de Software Brasileiro (MPS.BR), adaptado para a realidade e a cultura local e dentro dos padrões internacionais como ISO/IEC 12.207, ISSO/IEC 15.504 e CMMI-DEV. Trazendo mais credibilidade e segurança para os produtos de software desenvolvidos no Brasil. A seção 3.4.2 trará mais informações sobre o MPS.BR.

2.2 PERFIL DAS EMPRESAS

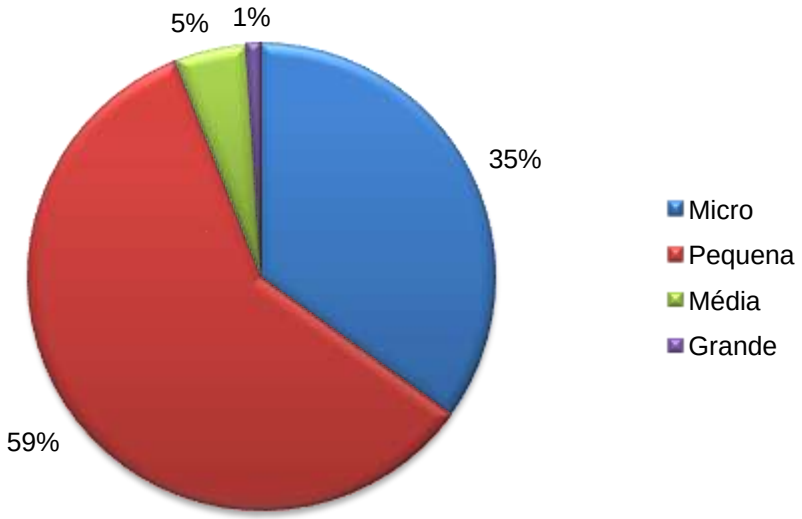

Gráfico 2.1 – Distribuição do mercado brasileiro de software de acordo com a classificação das empresas

Segundo a pesquisa da Abes (FILHO, HOCHSTETLER, *et al.*, 2006, p. 21), existiam cerca de 7,7 mil empresas que atuando no desenvolvimento, produção e distribuição de software e prestação de serviços. Deste total, 54% são de empresas de distribuição, 22% de serviços relacionados e 24% atuam na produção de software. Além disso, como mostra o Gráfico 2.1, as empresas que empregam menos de 50 funcionários (micro e pequenas) equivalem a 94% do mercado.

2.3 ORGANIZAÇÃO

A organização das micro e pequenas empresas de software do país é também algo muito importante e deve ser levado em consideração sob vários aspectos. O principal deles é a organização dos recursos humanos e dependendo de como acontece na prática, pode trazer sérios problemas que influenciam diretamente nas atividades. Esta seção fará uma explanação sobre como esses aspectos acontecem no dia-a-dia.

2.3.1 Colaboradores

Segundo o SEBRAE (2005), desde 1999 o critério adotado para conceituar micro e pequena empresa de comércio ou serviço é a receita bruta anual, ou o número de funcionários, cujos valores foram atualizados pelo Decreto nº 5.028/2004, de 31 de março de 2004, segundo descrito na Tabela 2.1. Ela não é um padrão nacional, serve apenas como referência[4] para as economias locais (Unidades da Federação).

Classificação	Renda bruta anual	Número de funcionários
Microempresa	Menor que 433.755,14	Menor ou igual a 9
Pequena empresa	Entre 433.755,14 e 2.133.222,00	Entre 10 e 49

Tabela 2.1 - Classificação das micro e pequenas empresas de software brasileiras segundo o SEBRAE

Observando o mercado local, é muito difícil para uma microempresa (tendo um quadro reduzido de pessoal) adotar a um processo de desenvolvimento de software, sem que uma pessoa não acumule papéis. Isso é comum no mercado. A maioria das empresas prefere dar mais papéis para os colaboradores do que contratar novos funcionários. Isso é uma característica observada e tratada pelo Prumo. Ele propõe a distribuição de papéis entre um número mínimo de pessoas, julgado através do interesse (artefatos e responsabilidades) do papel nas atividades do processo.

2.4 PROBLEMAS DA EMPRESA DE DESENVOLVIMENTO

O principal problema enfrentado pela empresa (ainda não aberta oficialmente) que desenvolve o Sistema para Gerenciamento de Crediários é a organização das atividades para o desenvolvimento e um bom produto de software. Assim como nas empresas brasileiras de software, ela possui conhecimento técnico sobre várias tecnologias. Dessa forma a equipe de desenvolvimento corre o risco de se confundir

[4] Cada um dos estados brasileiros reajusta a sua tabela de acordo com as características do mercado local.

na hora de desenvolver as atividades de produção do software, por que os integrantes dela não sabem em que momento deve-se especificar qual parte do software produzido e que atividade deve ser mais intensa para garantir que o problema do cliente será resolvido de fato.

Assim observa-se que o principal problema não está no conhecimento ou aprendizado da equipe, e sim, na sistematização das atividades que levam a construção de um produto de software viável para as duas pontas (empresa e cliente).

3 PROCESSO DE DESENVOLVIMENTO DE SOFTWARE

Esta seção mostrará em mais detalhes uma proposta para definição, nesse contexto, de processo (genérico) de software. Em seguida, complementa-se a ideia, mostrando os modelos mais citados pelas bibliografias consultadas e as mais usadas no dia-a-dia das empresas. Finalizando com alguns modelos, denominados aqui como: *padrões por excelência*, por serem considerados muito importantes para o desenvolvimento de um bom processo de software.

3.1 DEFINIÇÃO DE PROCESSO

Por ser um termo com ampla utilização nas diversas áreas do conhecimento, faz-se necessário um direcionamento do seu significado ao tema em questão. A primeira definição é abrangente o bastante para explicar o conceito, logo, um processo pode ser definido como uma série de ações na qual uma ou mais entradas são utilizadas para produzir uma ou mais saídas (AMBER, 1998). Característica bastante visível nas atividades do Prumo.

3.2 DEFINIÇÃO DE PROCESSO DE SOFTWARE

Assim, pensando em Engenharia de Software, define-se como um conjunto de passos parcialmente ordenados visando atingir uma meta que, neste caso, é entregar um produto de software com qualidade, dentro dos prazos e custos estabelecidos. Diversos autores renomados tentam definir processo de software enfatizando os aspectos mais importantes. Sendo assim, SOMMERVILLE (2001, p. 44) define como sendo um conjunto de atividades direcionadas à construção de um produto de software. Esta definição pode ser ainda mais abrangente aplicando as seguintes palavras de PRESSMAN:

> Um processo de software é base para o controle gerencial de projetos de software e estabelecimento do contexto no qual os métodos técnicos são aplicados, os produtos de trabalho (modelos, documentos, dados, relatórios formulários, etc.) são produzidos, os marcos são estabelecidos, a qualidade é assegurada e as modificações são adequadamente geridas (2006, p. 17).

Observa-se facilmente que todos eles possuem um mesmo referencial: atingir o objetivo de entregar um produto de software com qualidade e atendendo as expectativas do cliente. Quando o objetivo é alcançado, todos os envolvidos no processo estarão satisfeitos, tanto a equipe, por desenvolver um bom trabalho, quando o cliente, por ter um sistema confiável e financeiramente acessível.

3.3 MODELOS DE PROCESSOS

O estudo de modelos de processos de software é interessante para observar como a organização das atividades faz a diferença no momento de se produzir um software. Por isso, os tópicos seguintes abordam os modelos Cascata, Iterativo, Incremental, Espiral e traz exemplos de metodologias Ágeis de desenvolvimento, tentando explicar como eles abordam e organizam suas propostas de processos de software e apresentando seus pontos positivos e negativos.

3.3.1 Modelo em cascata

É o paradigma mais antigo da engenharia de software, sendo algumas vezes chamado de ciclo de vida clássico (PRESSMAN, 2006, p. 39). É um modelo de desenvolvimento de software sequencial no qual o desenvolvimento é visto como um fluir constante para frente (como uma cascata) através das fases de análise de definição dos requisitos, design do software, implementação e teste de unidade, integração e teste do sistema, operação e manutenção.

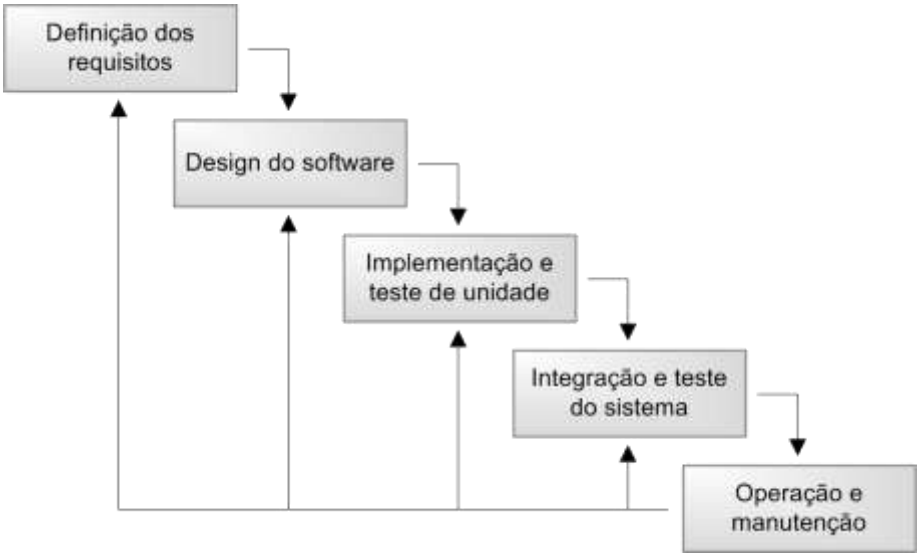

Figura 3.1 - Ciclo de vida do modelo sequencial (cascata)

A origem do termo *cascata* é frequentemente citado como sendo um artigo publicado em 1970 por W. W. Royce.

> Ironicamente, Royce defendia uma abordagem iterativa para o desenvolvimento de software e nem mesmo usou o termo *cascata*. Ele originalmente descreve o que é hoje conhecido como o modelo em *cascata* como um exemplo de um método que ele argumentava ser um risco e um convite para falhas (Wikipédia, A enciclopédia livre, 2009).

3.3.2 Modelo iterativo e incremental

Desenvolvimento incremental é uma estratégia de planejamento em que várias partes do sistema são desenvolvidas em versões, como na Figura 3.2, ou em paralelo, sendo integradas quando completas. Não implica, requer ou pressupõe desenvolvimento iterativo ou em cascata – ambos são estratégias de retrabalho. A alternativa ao desenvolvimento incremental é desenvolver todo o sistema com uma integração única.

Desenvolvimento iterativo é uma estratégia de planejamento de retrabalho em que o tempo de revisão e melhorias de partes do sistema é pré-definido. Isto não pressupõe desenvolvimento incremental, mas funciona muito bem com ele. Uma diferença típica é que a saída de um incremento não é necessariamente assunto de um refinamento futuro, e seu teste ou retorno do usuário não é utilizado como entrada para planos de revisão ou especificações para incrementos sucessivos. Ao contrario, a saída de uma iteração é examinada para modificação, e especialmente

para revisão dos objetivos das iterações sucessivas (Desenvolvimento Iterativo e Incremental, 2009).

3.3.3 Diferenciando os modelos em cascata, iterativo e incremental

Para firmar os conceitos dos modelos apresentados acima, foi esquematizada uma pequena comparação entre elas na Figura 3.2. Têm-se como exemplo, três partes que juntas compõem o software desejado. Assim, no modelo sequencial ou cascata, todos os software é trabalhado e desenvolvido de uma só vez.

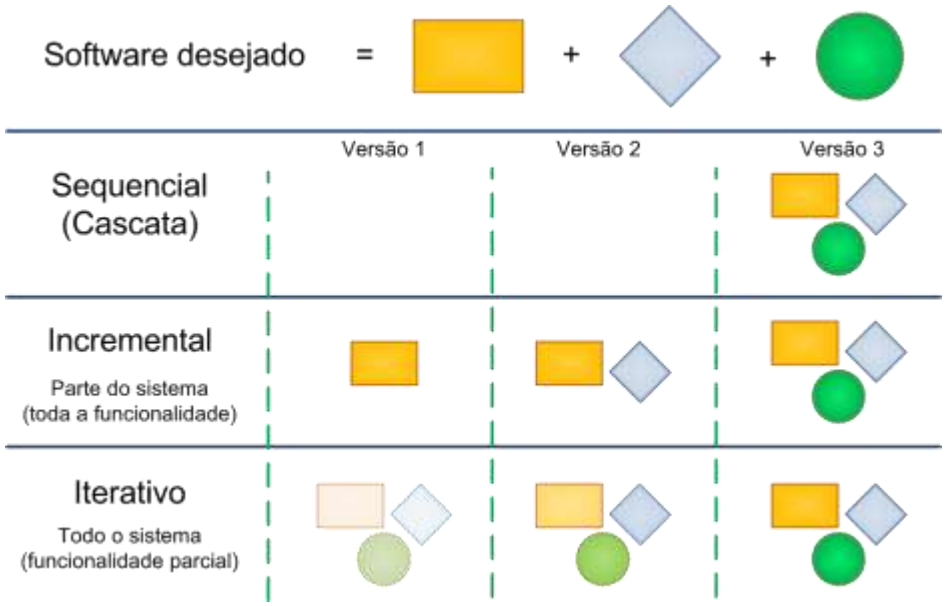

Figura 3.2 - Comparação entre os modelos em cascata, incremental e iterativo

Veja que o modelo iterativo lança versões de funcionalidades parcialmente implementadas. Na abordagem Incremental, são desenvolvidas as funcionalidades por completo, uma por uma, e lançadas incrementalmente em versões até que a última apresente as a solução desejada pelo cliente.

O desenvolvimento sequencial apresenta um aspecto que não é muito interessante para o mercado de software atual. Como se pode ver, o software é especificado, desenvolvido, testado e integrado de forma completa e implantado de uma só vez. Assim, os desenvolvedores não conseguem acompanhar as necessidades do cliente

com as regras de negócio dele mudando rapidamente, devido ao dinamismo do mercado. Sendo assim, ao praticar o desenvolvimento de software com essa abordagem, é provável que o ele não consiga atender as necessidades do cliente desde a sua concepção. Porém, esta forma de desenvolver software facilita o aprendizado e entendimento, pelo fato de que as atividades estão organizadas sequencialmente.

A vantagem do modelo incremental, em relação ao sequencial, é que ele possibilita a inserção de funcionalidades de cada vez. Assim, o cliente começa a utilizar uma parte da solução de cada vez, solicitando mais funcionalidades ou módulos para desenvolvimento e integração. A desvantagem desse modelo é que, uma vez definida a funcionalidade e o desenvolvimento é iniciado, o cliente só poderá solicitar alterações após o desenvolvimento dela. Podendo, em alguns casos, causar o desenvolvimento errado de uma funcionalidade, devido à uma má interpretação dos requisitos ou se o cliente não tiver se expressado da forma correta.

A iteratividade por si só já é interessante, ou seja, criar uma funcionalidade de forma iterada permite ao cliente expressar sua necessidade de forma fragmentada e analisada, observando os equívocos e mudanças ocorridas no ambiente do cliente. Isso torna o produto de software mais adequado às necessidades do cliente. Em contrapartida, essa iteração no desenvolvimento pode levar o cliente a atrapalhar o andamento do projeto, com a inserção desordenada de funcionalidades.

3.3.4 Modelo em espiral

Este modelo tem o objetivo de prover um “metamodelo” que pode acomodar diversos processos específicos. Isto significa que podemos encaixar nele as principais características dos modelos vistos anteriormente, adaptando-os a necessidades específicas de desenvolvedores ou às particularidades do software a ser desenvolvido. Este modelo prevê prototipação, desenvolvimento evolutivo e cíclico, e as principais atividades do modelo cascata.

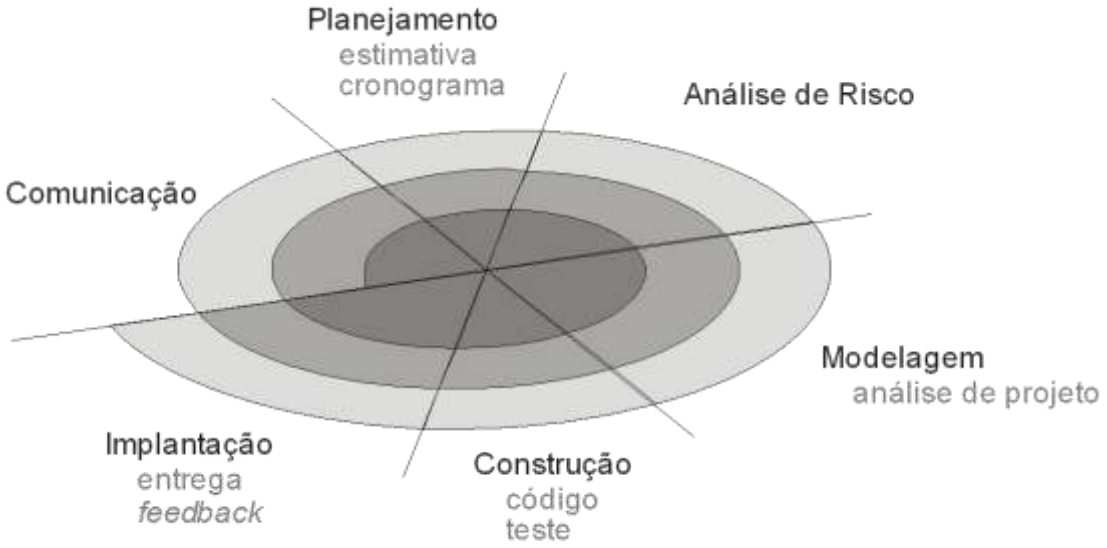

Figura 3.3 - Modelo de processo em espiral (PRESSMAN, 2006, P. 45).

Sobre esse modelo SIMÃO & VARELA dizem:

> O modelo em espiral foi proposto por Boehm em seu artigo de 1988 *A Spiral Model of Software Development and Enhancement*, como forma de integrar os diversos modelos existentes à época, eliminando suas dificuldades e explorando seus pontos fortes. Este modelo foi desenvolvido para abranger as melhores características tanto do ciclo de vida clássico como da prototipagem, acrescentando, ao mesmo tempo, um novo elemento - a análise de riscos - que falta a esses paradigmas (SIMÃO e VARELA, 2009, p. 10).

Um ponto a ser observado nesse modelo é a análise de risco que, segundo DUTRA (2008, p. 25), exige muita experiência da equipe. Além disso, pode ser difícil convencer os clientes que a abordagem evolucionária é controlável. Ela exige competência considerável na avaliação de riscos e depende dessa competência para ter sucesso. Se um risco importante não dor descoberto e gerenciado, fatalmente ocorrerão problemas.

3.3.5 Modelo Ágil

Os processos em desenvolvimento ágil de software parecem ser mais eficientes do que as metodologias antigas, porém, mesmo utilizando menos tempo do programador no desenvolvimento de softwares funcionais de alta qualidade, tem a desvantagem de ter uma perspectiva de negocio que não provê uma capacidade de

planejamento em longo prazo. Em essência, eles proveem mais funcionalidades por custo/benefício. Existem várias metodologias que podem ser consideradas como abordagens ágeis, entre elas: Scrum, Programação Extrema (XP), FDD, Crystal Clear, DSDM entre outras (Processo de Desenvolvimento de Software, 2009). As metodologias ágeis não serão definidas por estarem fora do contexto deste trabalho.

3.4 PADRÕES POR EXCELÊNCIA

Ao pesquisar sobre modelos e metodologias para desenvolvimento de software, você observará que existem inúmeras abordagens que exclamam serem as melhores, explicando os principais problemas que elas propõem resolver da melhor forma possível. Em contrapartida nem todas são adotadas pelas empresas e equipes de desenvolvimento. Algumas destas equipes até tentam usá-las, mas se a metodologia não for flexível o bastante ao ambiente onde está sendo empregada, então facilmente serão substituídas por outras que possuam essa características.

O termo “padrões por excelência” é usado aqui para denominar mecanismos e técnicas amplamente utilizados para o desenvolvimento de software, no Brasil e no mundo, respectivamente, o programa brasileiro MPS.BR e o *framework* RUP. Além de serem referenciadas no meio acadêmico, como modelos conceituais para o estudo da Engenharia de Software e suas aplicações.

O RUP permite que você customize facilmente um processo contendo unicamente as necessidades do seu projeto, permitindo a seleção e utilização apenas dos componentes necessários (International Business Machines - IBM, 2009). Por outro lado o MPS.BR é um programa que objetiva a melhoria do processo de desenvolvimento de software brasileiro. Ele não define um processo propriamente dito, mas sim o que um processo deve possuir. Para garantir que suas práticas estão sendo seguidas, são definidos guias de implementação e métodos de avaliação que usado por empresa “avaliadoras” contratadas para isso (MPS.BR, 2009, p. 2-6).

3.4.1 *Rational Unified Process* (RUP)

O *Rational Unified Process* é um processo de engenharia de software que oferece uma abordagem baseada em disciplinas para atribuir tarefas e responsabilidades dentro de uma organização de desenvolvimento. Sua meta é garantir a produção de software de alta qualidade que atenda às necessidades dos usuários dentro de um cronograma e de um orçamento previsível.

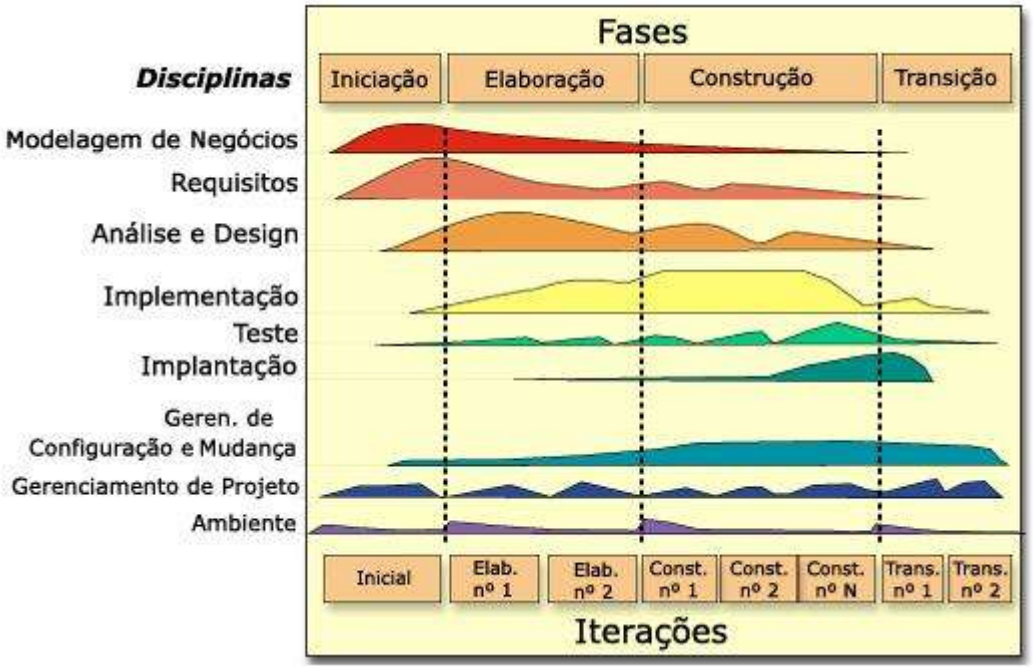

Figura 3.4 - Arquitetura organizacional do RUP

A Figura 3.1 mostra como a ênfase varia através do tempo. Por exemplo, nas iterações iniciais, é dedicado mais tempo aos requisitos. Já nas iterações posteriores, é gasto mais tempo com a implementação. Assim, pode-se observar que o RUP tem duas dimensões:

- **O eixo horizontal** - representa o tempo e mostra os aspectos do ciclo de vida do processo à medida que se desenvolve. Ele também representa o aspecto dinâmico do processo quando ele é aprovado e é expressa em termos de fases, iterações e marcos.
- **O eixo vertical** - representa as disciplinas, que agrupam as atividades de maneira lógica, por natureza. Representa o aspecto estático do processo,

como ele é descrito em termos de componentes, disciplinas, atividades, fluxos de trabalho, artefatos e papéis do processo

Este modelo de processo pode ser usado por uma “grande equipe” para desenvolver projetos de software robustos e complexos com uma grande variedade de funcionalidades, o que não é interessante aos objetivos deste trabalho. Porém, o RUP é flexível o bastante para quem empresas de tecnologia em busca melhorias e padronização nos seus processos de software molde o seu próprio processo, utilizando os modelos sugeridos e adaptando as atividades e a realidade do mercado e da equipe.

3.4.2 MPS.Br

É um programa mobilizador, de longo prazo, criado em dezembro de 2003 e coordenado pela SOFTEX, que conta com apoio do Ministério da Ciência e Tecnologia, FINEP, SEBRAE e Banco Interamericano de Desenvolvimento (SOFTEX, 2009).

O objetivo do programa MPS.BR (acrônimo) é a Melhoria de Processo do Software Brasileiro, com duas metas a alcançar a médio e longo prazos:

- **Meta técnica** - visando à criação e aprimoramento do modelo MPS, com resultados esperados tais como:
    - Guias do modelo MPS;
    - Instituições Implementadoras credenciadas para prestar serviços de consultoria de implementação do modelo de referência MR-MPS;
    - Instituições Avaliadoras credenciadas para prestar serviços de avaliação seguindo o método de avaliação MA-MPS;
    - Consultores de Aquisição certificados para prestar serviços de consultoria de aquisição de software e serviços relacionados;
- **Meta de mercado -** visando à disseminação e adoção do modelo MPS, em todas as regiões do país, em um intervalo de tempo justo, a um custo razoável, tanto em PME (foco principal) quanto em grandes organizações públicas e privadas, com resultados esperados tais como:
    - Criação e aprimoramento do modelo de negócio MN-MPS;

- Cursos, provas e workshops;
- Organizações que implementaram o modelo MPS;
- Organizações com avaliação MPS publicadas (prazo de validade de três anos).

4 O PRUMO

Como é parte dos objetivos desse trabalho, esta seção pretende apresentar uma proposta de processo de desenvolvimento de softwares para utilização em micro e pequenas empresas tecnologia da informação. Antes que seus detalhes sejam citados, uma forma de visualização do processo é apresentada com o intuito de deixar o leitor bem posicionado em termos de qual característica e nível de detalhamento ele está observando, além de proporcionar um panorama completo da organização do Prumo. Outra estratégia adotada para melhorar o entendimento do conteúdo apresentado aqui foi a definição dos papéis antes que as atividades sejam apresentadas e detalhadas. Assim o leitor compreenderá melhor o objetivo de cada uma delas e, entenderá a escolha dos papéis, dos artefatos necessários para a realização da atividade e dos artefatos gerados como saída.

4.1 NÍVEIS DE DETALHAMENTO DO PROCESSO

Um processo de desenvolvimento de software pode ser observado sobre diversos aspectos. Uma forma de abordá-lo (a fim de entender a sua organização) é buscando uma visão em níveis de detalhamento. Pensando nisso, esse tópico, tenta explicar o Prumo em três níveis: Ciclo de vida, Atividades e Detalhamento, mostrados na Figura 4.1.

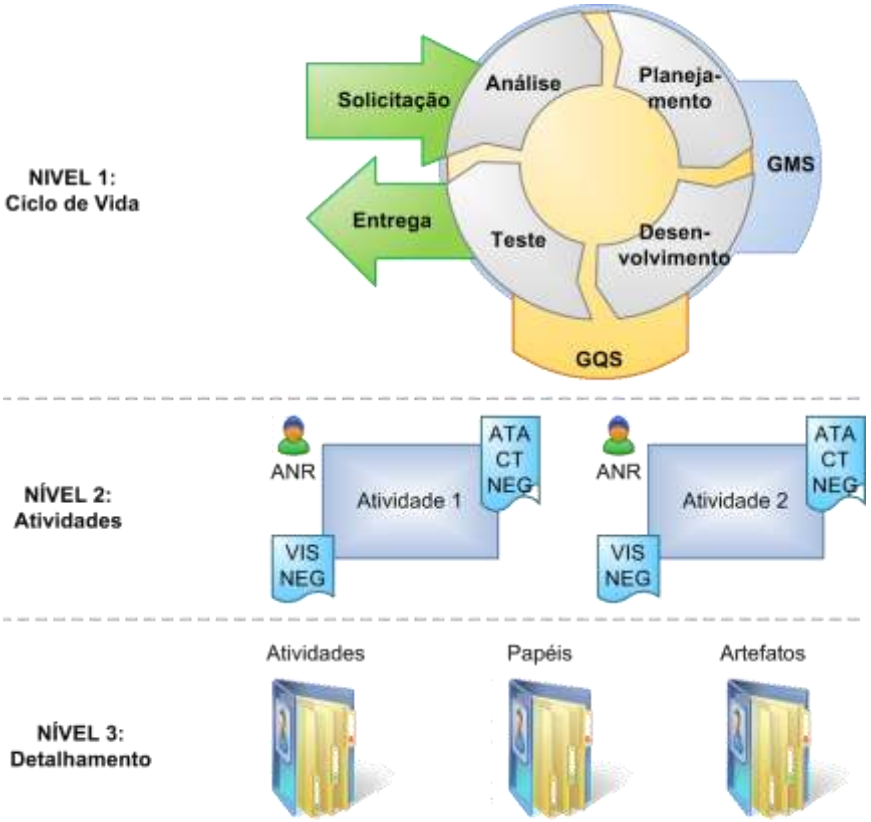

Figura 4.1 - Visualização do Prumo em níveis

No nível mais alto de abstração dos detalhes está o Ciclo de Vida do processo, onde se especifica como estão organizadas as fases. Durante a definição deste nível, pensou-se como seriam melhor utilizados dos modelos de processos definidos pela Engenharia de Software. Note que foram aplicados aspectos do modelo iterativo e incremental ao Prumo, ou seja, no ciclo podem ser "lançados" pacotes de funcionalidades que após serem desenvolvidos e integrados representam uma solução.

O seguindo nível possui a representação[5] das atividades compostas de: nome da atividade, participantes, artefatos de entrada e artefatos de saída. Outra característica importante expressa nesse nível é o relacionamento entre as atividades. Assim os integrantes da equipe sabem exatamente em que atividades eles possuem participação, quais artefatos necessários para iniciar a atividade e quais artefatos gerados como saída.

Por fim, no terceiro nível, estão todos dos detalhamentos necessários para a execução bem sucedida do processo. São eles:

[5] No Apêndice A – Ciclo de vida e atividades, você encontrará o detalhamento deste nível.

- **Detalhamento das atividades** – para cada uma delas é definida uma visão geral, papéis responsáveis e participantes, artefatos de entrada e saída e uma visão detalhada, contendo passos principais e alternativos;
- **Definição dos papéis** – para cada um deles é realizada uma breve descrição, listadas as atribuições, artefatos que são de responsabilidade do papel, títulos exigidos, títulos desejados e as características pessoais esperadas da pessoa candidata ao papel;
- **Definição dos artefatos** – nesse nível, são disponibilizados os modelos dos artefatos necessários para controle e execução do processo. A estrutura organizacional deles pode variar de acordo com o objetivo.

É importante frisar que qualquer micro e pequena empresa de software pode adotar o processo e tentar adaptá-lo de acordo com as suas necessidade, pressupondo-se que tais adaptações serão observadas por um profissional com conhecimentos seguros em Engenharia de Software e nas práticas do RUP e MPS.BR.

4.2 NÍVEL DO CICLO DE VIDA

Para dar início a este tópico é importante ressaltar a diferença entre *ciclo de vida de processo* e *ciclo de vida de software*. São dois conceitos bem parecidos e que causam confusão em muitas pessoas. Assim, o *ciclo de vida de um processo* descreve as fases e atividades pelas quais um projeto de software passa até ser gerada uma nova funcionalidade ou solução, a partir de uma necessidade expressa em forma de solicitação. Diferente dessa definição, o *ciclo de vida de um software* inicia a partir da concepção do software, passando por um processo de desenvolvimento, seguindo pela sua utilização e evolução até que ele não atenda as necessidades do seu utilizador e caia em desuso. Tomando como base esses conceito, foi definido o Prumo dividido em fases do processo principal e processos de apoio.

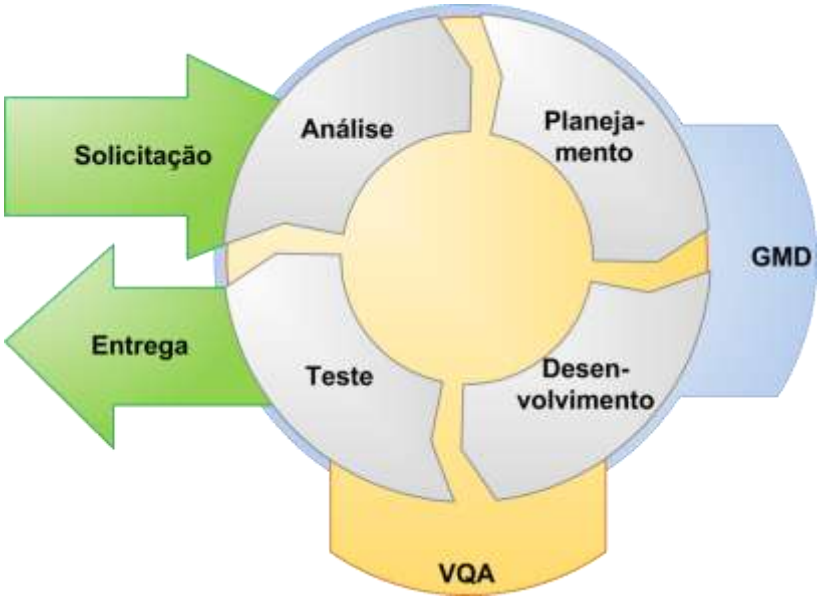

Figura 4.2 - Ciclo de vida do Prumo

Assim como podemos observar na Figura 4.2, o Prumo foi definido e separado em fases que descrevem as entradas e saídas do ciclo de desenvolvimento (Solicitação e Entrega), o próprio ciclo (Análise, Planejamento, Desenvolvimento e Teste) e processos de apoio que são o acompanhamento para gerência da qualidade (VQA) e a gerência de mudanças (GMD). Cada uma delas abordando o desenvolvimento do software sobre uma visão diferente de acordo com os seus interesses. A seguir, cada uma das fases serão explicadas para que haja maior compreensão das suas abordagens e características.

Antes de iniciar as descrições das fases do ciclo de vida do Prumo, será realizada uma breve definição de do que significa o termo *fase* dentro do escopo deste trabalho. Assim, *fase* é a representação de um conjunto de atividades organizadas com um objetivo comum de desenvolver uma parte significativa de um projeto de software.

Geralmente elas são dependentes entre si em alguns aspectos. Os artefatos são um exemplo, ou seja, numa *fase*, geralmente são usados artefatos criados e desenvolvidos em outras fases. Por exemplo: o Documento de Elicitação de Requisitos é criado na fase de análise e utilizado nas fases de planejamento e desenvolvimento.

4.2.1 Fase de solicitação

Como observada na Figura 4.2, esta é a fase de entrada no processo. Ela descreve como o cliente pode expressar suas necessidades perante a equipe de desenvolvimento e como eles devem observar o problema em busca da melhor solução. Em linhas gerais, o objetivo dessa fase é identificar o problema, entender o ambiente do cliente e lançar uma proposta de solução. Então a solução só será encaminhada para a próxima fase depois de autorizado pelo cliente e o mesmo assinar um contrato prévio.

4.2.2 Fase de análise

Aqui serão utilizados os meios mais apropriados da Engenharia de Software para especificar as necessidades do cliente em forma de requisitos e casos de uso, em um nível de detalhamento aceitável para a equipe de desenvolvimento e para o projeto. Para chegar a esse nível de aceitação, são criados vários tipos de especificações (incluindo os diagramas e protótipos) que descrevem as necessidades do cliente e a solução proposta pela equipe de desenvolvimento. Ao final desta fase, estarão prontas todas as especificações dos requisitos, ou seja, o problema do cliente estará totalmente entendido e a solução modelada.

Uma característica muito importante dessa fase é a constante participação do cliente na maioria das atividades, na tentativa de garantir um melhor entendimento dos problemas e necessidades dele. Dessa forma, o cliente estará interagindo com a equipe de desenvolvimento, tirando dúvidas, descrevendo necessidades e ajudando a moldar a solução. Essa é uma estratégia que tenta garantir o aumento da interação e comunicação entre o cliente e a empresa, criando uma relação mais fortalecida e diminuindo a sensibilidade do cliente perante as renegociações de prazo e custo (havendo necessidade).

4.2.3 Fase de planejamento

A fase de planejamento é uma fase de preparação e gerenciamento, onde são definidos os planos para execução das próximas fases. Nesta fase, é necessário avaliar o ambiente de desenvolvimento disponível para o desenvolvimento da

solução. Dependendo das exigências do projeto e do cliente ou até mesmo dos requisitos, o ambiente deve ser ajustado para acomodar satisfatoriamente a construção da solução. Durante a elaboração de um Plano de Projeto[6], devem ser levados em consideração vários aspectos:

- Observar a *capacidade da equipe* em trabalhar com a tecnologia necessária para desenvolver a solução;
- Analisar a *necessidade de treinar a equipe* ou contratar pessoal já qualificado, para isso, fazendo um estudo e verificando a viabilidade de custo e prazo das opções disponíveis;
- Organizar o ambiente da empresa, *alocando recursos humanos e materiais* de forma que possibilite maior comodidade e agilidade da equipe de desenvolvimento.

É importante observar e ponderar outras características que podem surgir em cada ambiente onde o processo é empregado ou em cada tipo de solução desenvolvida. Além disso, é necessário identificar todos os riscos que podem vir a modificar o andamento do projeto em questão e tomar as medidas cabíveis para evitá-los ou gerenciá-los de maneira que o projeto sofra menos impacto.

4.2.4 Fase de Desenvolvimento

No ciclo de vida do Prumo, esta corresponde à terceira fase. Aqui serão definidas as especificações arquiteturais, banco de dados e codificação da solução. Um exemplo de especificação arquitetural é: atualmente a maioria dos projetos são definidos seguindo o paradigma de Programação Orientada a Objetos (POO), assim sua arquitetura é constituída de classes e organizada em pacotes de funcionalidades. Além disso, são criados vários diagramas utilizando UML para mostrar o relacionamento entre as classes dentro dos pacotes.

Outra especificação importante nesta fase é a definição do esquema de banco de dados para a solução, sendo, para isso, desenvolvidos Diagramas Endidade-Relacionamento (DER), em seguida os Diagramas Entidades-Relacionais (ER) e por

[6] Consulte o Apêndice C para saber os detalhes o Plano de Projeto e outros artefatos.

fim, definição de um dicionário de dados. Estes podem fazer parte de um novo banco de dados ou serem integrados a outro.

A terceira e última atividade desta fase é a codificação, é nesse passo que são traduzidas todas as especificações arquiteturais e de banco de dados para uma linguagem de programação, resultando no código fonte ou *build* da solução.

4.2.5 Fase de Teste

Esta fase é responsável por definir como serão realizados todos os testes necessários para garantir o bom funcionamento e aceitação da nova solução de software, que poderá ser integrada a outra solução existente ou um novo software. Alguns dos testes propostos são descritos abaixo:

- Teste de integridades dos dados;
- Teste de interface com o usuário;
- Teste de segurança;
- Teste de funcionamento;
- Teste de aceitação;
- Teste de integração;

De forma geral, após a realização de todos os testes sobre as funcionalidades ou soluções, em caso de aprovação, elas devem seguir para a próxima fase, caso sejam identificadas irregularidades, serão encaminhadas (juntamente com relatório de irregularidades) para a fase e atividade responsável por corrigir o erro detectado. Para complementar o entendimento sobre as atividades desta fase, o leitor pode consultar os Apêndice A – Ciclo de vida e mapa de atividades.

4.2.6 Fase de Entrega

Esta fase, como explicado na seção 4.2, é a fase de saída da funcionalidade do ciclo de vida do processo. Nela são desenvolvidos os planejamentos para treinamento dos usuário e implantação da solução. Os cursos preparatórios para que os usuários aprendam a utilizar a solução antes da utilização serão desenvolvidos. Após o

treinamento, a nova solução já poderá ser implantada, podendo ser necessário um acompanhamento da utilização durante alguns dias, caso o cliente solicite.

Nessa fase o acompanhamento é importante para que se tome conhecimento de como a solução está sendo encarada na prática. Além disso, é importante observar qual o comportamento dos usuários em relação à nova funcionalidade e se esta está desempenhando satisfatoriamente sua função.

4.2.7 Verificação da qualidade

A fase de verificação da qualidade ou validação faz parte de um dos processos de apoio do Prumo, sendo realizado sempre nas transições entre as fases principais do ciclo de vida do processo. O objetivo dele é garantir a qualidade do produto de software que está sendo criado, sendo necessária, para isso, uma verificação na qualidade dos principais produtos de software criados. Além disso, a verificação também acompanha a qualidade do processo como um todo a partir de uma análise da quantidade de não-conformidades em comparação com o cumprimento dos prazos estabelecidos para o projeto. Logo após, as devidas correções são apresentadas aos responsáveis e a atividade segue o curso normal.

Os papéis responsáveis pelas atividades durante esta fase são o Gerente de Projeto e Analista de Qualidade. O primeiro irá avaliar como esta o andamento do projeto, se está em dia, se os ricos estão sendo controlados devidamente, etc. O segundo irá avaliar os artefatos gerados durante a fase, de forma a solicitar alterações nos mesmos, caso eles não estejam no padrão estabelecido pela empresa.

Em seguida haverá uma reunião entre os dois, onde o Analista de Qualidade passará ao Gerente a sua avaliação, para que ele possa ter um maior controle do andamento do projeto, já que pode haver a necessidade de correção de alguns documentos.

4.2.8 Gerenciamento de mudança

Este é um processo de apoio que será realizado quando qualquer envolvido solicitar uma mudança no projeto, uma vez que esta mudança envolva uma especificação que já tenha passado pela Fase de Análise. O comportamento dessa atividade foi

definido dessa forma devido ao impacto que uma mudança pode provocar em um item em desenvolvimento.

Na primeira atividade será analisada a influência que a mudança solicitada irá gerar no projeto, analisando se algum requisito será invalidado ou adicionado ao projeto pela solicitação ou ainda se o processo terá que parar o desenvolvimento de alguma funcionalidade. Uma vez mensurados os impactos da solicitação serão avaliados e os gastos referentes à mudança serão calculados, segundo horas trabalhadas ou utilizando alguma outra técnica interessante que traga agilidade e precisão na mensuração do esforço da equipe.

Em seguida haverá uma reunião com o Cliente onde serão apresentados os impactos analisados. Depois o Gerente de Projetos, junto com o Analista de Requisitos, irá negociar a viabilidade das alterações com o Cliente sugerindo alternativas e soluções. Logo após, serão definidas as adições, mudanças ou cancelamentos de funcionalidades ou do projeto como um todo.

Por último, será executada a mudança definida na reunião com o Cliente. Caso o ele decida que o projeto seja cancelado, serão apresentados a ele todos os encargos especificados no contrato. Caso a solicitação consista em uma mudança no projeto, os documentos serão atualizados e a equipe será informada. A mudança solicitada será analisada com o apoio, principalmente, da Matriz de Rastreabilidade que identificará quais os impactos que tal mudança provocará no projeto. Eles podem ser excluídos atualizados ou complementados e acoplados ao projeto, então será gerada uma nova solicitação comum (a partir da fase de solicitação) e o processo seguirá naturalmente.

4.3 OS PAPÉIS, RESPONSABILIDADES E ARTEFATOS

Para que uma empresa possua uma organização de atividades eficiente, ela precisa definir “quem faz o que”. Por isso, antes da apresentação das atividades, todos os papéis serão definidos, expondo suas responsabilidades, atribuições e apontando suas características mais importantes.

É importante frisar nesse momento sobre a terminologia utilizada nas próximas seções para denominar os papéis e artefatos definidos aqui. Assim, quando forem citados nomes de papéis ou artefatos, eles terão a primeira letra escrita em maiúsculo, por exemplo: a diferença entre “Cliente” e “cliente” é que o primeiro faz referência o papel definido no Prumo e o segundo representa uma pessoa qualquer que procura uma empresa em busca de soluções (produtos ou serviços).

4.3.1 Cliente

Este papel representa a pessoa que entende a regra de negócio da empresa a qual se destina a solução podendo ser representado por uma ou mais pessoas, dependendo do projeto e da necessidade. Nesse caso ele pode ser dividido em dois perfis: *Cliente Interno* e *Cliente Externo*, onde cada uma das pessoas apresentará características e responsabilidades pertencentes a um ou outro perfil. Em todos os casos eles possuirão as mesmas atribuições.

O primeiro perfil analisado é o *Cliente Interno* representado por um colaborador da empresa de desenvolvimento. Geralmente este perfil pode possuir poucos conhecimentos técnicos em informática, porém possui boas habilidades para entender regras de processos de negócios de outras empresas e sabe persuadir o cliente em busca de descrições mais precisas das necessidades dele. Esse perfil se responsabiliza pelo desenvolvimento de alguns artefatos do projeto.

- O primeiro artefato é o Glossário[7], que é um documento onde estão reunidos todos os termos incomuns, tanto para o cliente quanto para a equipe. O objetivo dele é definir um vocabulário compartilhado entre os *stakeholders* (envolvidos) do projeto, a fim de unificar o entendimento para que não haja ambiguidades. No Prumo ele é usado para especificar, de forma unificada, os termos comuns ao Cliente e a equipe. Pode ser dividido em: Glossário de Negócios (termos comuns aos negócios do Cliente) e Glossário de Projeto (termos comuns à equipe do projeto).
- O outro artefato é denominado como Regras de Negócio. Nele o *Cliente Interno* especifica detalhadamente todas as regras de negócio, e que fazem parte do domínio do problema. Para a especificação poderiam ser usados

[7] Além do Glossário, outros artefatos estão detalhados no Apêndice C.

cenários[8] de uso. Este é um documento muito importante para a compreensão do problema e principalmente para o controle das mudanças, já que, na maioria das vezes, o ambiente do cliente é dinâmico e suas regras de negocio mudam com facilidade.

O segundo perfil é o do *Cliente Externo* incorporado por uma pessoa externa à empresa de desenvolvimento. Geralmente representado pela pessoa que procura a solução ou alguém indicado por ele com capacidade para responder às questões problemáticas enfrentadas. Este perfil de cliente pode apresentar pouco ou quase nenhum conhecimento sobre computadores de um modo geral.

- Sobre artefatos, o *Cliente Interno* é responsável por todos os documentos fornecidos por ele para definição e entendimento do problema. Esses artefatos podem ser dos mais variados tipos: relatórios, planilhas, anotações, desenhos, documentos, etc.

Até o momento foram analisados esses dois perfis de Cliente que se encaixam no Prumo, porém nada impede que outros perfis sejam observados e considerados.

4.3.2 Analista de Requisitos

O papel de Analista de Requisitos representa a pessoa que consegue observar o ambiente do Cliente identificando os problemas e propondo soluções viáveis e interessantes para o ele. O Analista de Requisitos também deve possuir habilidades para persuadir o Cliente, deixando-o a vontade para expor suas ideias e necessidades, independente do perfil de Cliente considerado.

Como observado no Apêndice B.2 (definição do papel) e Apêndice A (ciclo de vida e mapa de atividades), o Analista de Requisitos possui atribuições de comunicação com o Cliente e desempenha atividades importantes para a empresa, por isso, ele é responsável pelo artefato Documento de Visão de Negócio. Neste documento estão contidas informações relevantes, como as vantagens que o cliente terá caso ele contrate a solução. Além disso, esse documento especifica em detalhes o ambiente do cliente e identifica quais os possíveis usuários e envolvidos no projeto, mostrando

[8] Um cenário é uma narrativa, textual ou pictórica, de uma situação (de uso de uma aplicação), envolvendo usuários, processos e dados reais ou potenciais (REZENDE, 2003).

para cada um deles, as características, atribuições e necessidades. Adicionalmente, são elicitados todos os problemas enfrentados e as possíveis soluções para eles.

Outra atividade desempenhada pelo Analista de Requisitos é a identificação, definição e especificação de requisitos em casos de uso. Em consequência disso, ele se torna responsável pelos artefatos Documento de Elicitação de Requisitos e Documento de Especificação em Casos de Uso.

No Documento de Elicitação de Requisitos, ele organiza os requisitos, dando um número identificador e uma descrição para cada um dos requisitos funcionais (RF – representam as funcionalidades da solução), não funcionais (RNF – representam as características da solução) e cria um escopo negativo, deixando explícito o que não faz parte da solução.

Uma forma de ele organizar os requisitos e identificar o relacionamento entre eles é criando uma Matriz de Rastreabilidade. Este artefato pode relacionar tanto RF com RF, quanto RF com RNF, sendo importante para identificar o impacto nas alterações dos requisitos. Ele também pode relacionar várias outras características de um projeto através da rastreabilidade entre elas, ou seja, a interligação existente entre características e especificações da solução. Para realizar esse controle, podem ser usadas planilhas eletrônicas, onde na linha estão expostas uma das características, e na coluna a outra. Para ter uma visão mais concreta sobre como seria esse tipo de controle, consulte o Apêndice C.9, nele há um exemplo de uma Matriz de Rastreabilidade que pode ser tomada como base para a construção sua própria matriz.

Por fim, no artefato Documento de Especificação de Casos de Uso, o Analista de Requisitos cria diagramas de casos de uso UML, demonstrando claramente quem (atores) interage com que funcionalidade (caso de uso). e detalhando-os através de descrição formal ou diagrama de atividades UML. Aqui o ator representa qualquer entidade que interaja com o sistema informado dados. Ele pode ser um usuário, um subsistema ou até mesmo um sistema externo (Rational Software Corporation, 2002)..

4.3.3 Gerente de Projetos

Uma pessoa que detém as atribuições desse papel deve possuir espírito de liderança e trabalho em equipe. Um Gerente de Projetos deve ser capaz de liderar uma equipe de projetos de forma que ela venha ter o máximo de desempenho possível. Ele deve ser capaz de gerenciar de maneira efetiva os riscos de um projeto. Para isso ele deve possuir uma lista de possíveis riscos e junto a eles, um plano de ação preventiva ou corretiva, caso um deles venha a se concretizar, chamado *plano de contingência*.

Esta lista de riscos pode ser organizada da forma mais conveniente possível, de acordo com as considerações do Gerente de Projetos. Um exemplo seria organizá-la em um único documento denominado Lista de Riscos contendo os riscos e suas principais características. O documento Gestão de Ações Corretivas ajuda no controle e nas medidas a serem tomadas nos desvios que acontecem durante o projeto, além de possibilitar que tais desvios possam ser evitados nas iterações futuras.

Outro artefato que também é de responsabilidade deste papel são os Contratos. Assim, um Gerente de Projetos também interage com o Cliente, realizando negociações e firmando os contratos. O Prumo não define nenhum modelo de contrato. Portanto, devem ser definidos contratos de acordo com o serviço a ser prestado ao cliente.

Em se tratando de experiência, não é necessário que ele possua conhecimento técnico na tecnologia trabalhada, embora isto adicione certa importância. Ele deve ser líder, realizar planejamentos concretos e acompanhar o trabalho e o projeto.

4.3.4 Arquiteto de Software

Ao papel Arquiteto de Software são atribuídas a liderança e a coordenação das atividades técnicas no decorrer do projeto. O Arquiteto de Software também estabelece a estrutura geral de cada visão arquitetural do software: a decomposição da visão, o agrupamento dos elementos e as interfaces entre esses principais agrupamentos. Portanto, comparado aos outros papéis, a visão do Arquiteto de Software é ampla, e não detalhada (Rational Software Corporation, 2002).

Um dos artefatos pelo qual o Arquiteto de Software é responsável é o Documento de Arquitetura do Software que oferece diversas visões arquiteturais de representação da solução. Nele são inseridos, por exemplo:

- **Visão lógica** – representação do ponto de vista da arquitetura do modelo de *design,* dividindo em subsistemas e pacotes. Dentro de cada pacote está a representação das classes significativas do ponto de vista da arquitetura.
- **Visão de implantação** – descreve uma ou mais configurações físicas do computador ou da rede, indicando como o software utilizará esta estrutura.
- **Visão de dados** – descreve a perspectiva de armazenamento dos dados persistentes, podendo ser inseridos os diagramas DER e ER.

No Prumo o Arquiteto de Software também determina como será feita a integração de uma nova funcionalidade ao sistema, definindo qual a ordem em que devem ser desenvolvidos os *builds*[9] e quais os subsistemas que eles fazem parte. Para isso ele cria o artefato Plano de Integração do *Build*,

4.3.5 Desenvolvedor

Este papel representa a pessoa capaz de traduzir problemas do “mundo humano” para o “mundo computacional” através de linguagens de programação. O desenvolvedor deve ter conhecimentos sólidos sobre os princípios básicos da computação, como algoritmos e estruturas de dados, além de saber aplicá-los em uma ou mais linguagens de programação. O desenvolvedor dever saber trabalhar em equipe dividir o conhecimento com os demais colegas de trabalho.

Adicionalmente, ele participa do processo de mensuração dos requisitos e identificação dos possíveis riscos de implementação das funcionalidades e comenta o código fonte da solução, tornando-o de fácil compreensão.

Em se tratando de artefato, o Desenvolvedor fica responsável pelo Componente de Software que nada mais é, neste contexto, do que a codificação do modelo arquitetural lógico da solução, definido anteriormente pelo Arquiteto de Software.

[9] Versão codificada da solução e que ainda não foi submetido a nenhum teste.

4.3.6 Analista de Testes

O papel de Analista de Testes é representado por uma pessoa com habilidades para realizar diferentes tipos de testes sobre software. Durante suas atividades no ciclo de vida do processo, ele deverá atuar em duas fases distintas. A primeira atuação do Analista de Testes é na fase de análise, aonde serão realizados planejamentos para a realização dos testes nas fases seguintes. Durante o planejamento ele elabora um artefato denominado Plano de Testes que contém todas as diretrizes e resultados esperados para aqueles requisitos. O objetivo disso é garantir que as funcionalidades especificadas pelo Cliente não perderão a essência dos resultados esperados.

A segunda atuação deste papel é na fase de Testes. Nela o Analista de Testes acompanha a execução de todos os testes e compara os resultados com as diretrizes contidas no artefato Plano de Testes, definido anteriormente por ele.

Ele também realiza as atividades de um testador executando todos os testes especificados. Como atribuições, este papel desempenha, por exemplo: testes de unidade, testes de integração, testes de segurança de dados, testes de aceitação e demais testes necessários para garantir um bom nível de qualidade para a solução.

4.3.7 Analista de Suporte

O papel do Analista de Suporte deve ser incorporado por uma pessoa capaz de realizar treinamentos para os usuários sobre sistemas computacionais. Para realizar suas atribuições de maneira eficiente, ele deve ser comunicativo e ter boa relação interpessoal, já que ele estará em contato tanto com os futuros usuários da solução e com o Cliente.

O Treinador, antes da realização dos treinamentos, desenvolve um Plano de Treinamento, contendo todas as especificações necessárias para a realização dos treinamentos dos usuários. Nele estão contidas informações como: período de execução das atividades, produtos a serem treinados, recursos a serem utilizados, etc.

Durante a definição do Prumo foi decidido que o Treinador também realizará a implantação da solução. Para isso, ele deve realizar um Plano de Implantação com características bem parecida com as de um Plano de Treinamento, porém possui uma maior ênfase nas especificações de configuração de hardware e software.

Ele também é responsável pela emissão do Comprovante de Implantação da funcionalidade ou solução. Este artefato não possui um modelo definido aqui, ficando sua a criação a cargo da empresa de desenvolvimento que aderir ao processo,

4.3.8 Analista de Qualidade

Este papel é representado pela pessoa que realizará as auditorias sobre os artefatos produzidos durante o ciclo de vida do processo. Ele observa os artefatos, examinando-os segundo os padrões estabelecidos pela empresa de desenvolvimento.

Para melhor desenvolver suas atividades o Analista de Qualidade faz uso da Lista de Verificação da Qualidade. Ela funciona como um *checklist* possibilitando um maior controle do que foi desenvolvido ou não. Os documentos já criados e verificados são classificados em:

- **Conforme** – quando um artefato está, por completo, aderente às políticas de qualidade estabelecidas pela empresa;
- **Não-conforme** – quando um artefato ou parte dele, não obedece às políticas de qualidade da empresa;
- **Não se aplica** – quando não se pode classificar um artefato, parte dele ou outra característica, dentro dos padrões de qualidade estabelecidos;
- **Pendente** – quando uma não-conformidade já foi apontada em um artefato e, em uma segunda verificação, ela ainda permanece inalterada ou não apresenta qualidade satisfatória.

conforme, não-conforme, não se aplica e pendente.

Quando um dos itens de verificação é classificado como não-conforme, o Analista de Qualidade notificará ao papel responsável para que este possa realizar as devidas correções.

4.4 CONFROTAMENTO DE PAPÉIS

Como sendo um dos objetivos pretendidos para este trabalho, o Prumo também pode flexibilizar os seus papéis de modo a atender as necessidades do ambiente onde ele está sendo aplicado.

Sabe-se que muitas empresas de software, geralmente, trabalham com um quadro de funcionários inferior às suas necessidades, quando se trata de atividades. Ou seja, é observável que na maioria dos casos uma pessoa acumule vários papéis dentro de uma organização. Isso, não é encarado aqui como um problema, mas sim como característica, devido ao dinamismo das empresas modernas que exigem dos seus colaboradores conhecimentos e habilidades um tanto “distribuídas”. Neste caso, entenda como sendo uma pessoa que possui conhecimento em várias áreas e habilidade para lidar com diversas situações diferentes.

4.4.1 Distribuindo papéis em uma microempresa

Para tornar mais claro o entendimento a cerca dessa característica, considere um exemplo de uma empresa contendo cinco pessoas. Os papeis definidos para o Prumo serão distribuídos para esse número de pessoas, considerando a grande quantidade de atividades que cada papel possui, não tornando um ou outro papel sobrecarregado ao ponto de deixar o processo ineficiente.

Nos projetos de software cada papel tem seu objetivo e interesses definidos, porém alguns deles não podem ser exercidos simultaneamente pela mesma pessoa, por correrem o risco de agredir a qualidade do produto e do processo. Em contrapartida, outros papéis podem ser agrupados de forma eficiente devido às características que cada um deles possuem. A seguir continuaremos com um exemplo.

O primeiro agrupamento a ser considerado realiza a junção entre os papéis de Cliente (interno) e Analista de Requisitos, onde o segundo incorpora as atribuições

do primeiro. O objetivo dessa junção está no fato de o Analista de Requisitos também possuir características de comunição.com o Cliente (externo). Logo, o Analista de Requisitos passa a ser responsável pelo artefato de Regras de Negócio e Glossário.

O segundo agrupamento une o Arquiteto de Software com o Desenvolvedor. Nesse caso, o primeiro papel passará a desempenhar todas as atribuições do segundo, ficando responsável pelo Componente de Software e realizando todo o processo de codificação.

O terceiro agrupa o Gerente de Projetos com o Analista de Qualidade, sendo motivado pelo fato de o primeiro papel acompanhar todas as atividades do processo. Assim, além dele gerenciar a equipe, também terá melhores condições de gerenciar a qualidade dos artefatos produzidos.

A quarta forma de agrupamento proposta neste trabalho une dois papéis, o Analista de Testes e o Analista de Suporte, ou seja, agora o primeiro papel, fará, não só, o planejamento, definição de diretrizes de testes e execução, como também passará a realizar o planejamento e execução dos treinamentos e implantação da solução.

Qualquer outra forma de combinação entre papéis deve ser avaliada, por exemplo: os papéis de Desenvolvedor e Testador, não devem ser atribuídos a uma mesma pessoa devido ao risco de o Desenvolvedor criar testes ineficientes para os componentes de software que ele mesmo desenvolveu, entre outros casos.

4.5 NÍVEL DAS ATIVIDADES

As atividades dentro de um processo de desenvolvimento de software fazem parte do segundo nível de detalhamento de um processo. Elas são importantes para especificar em que momento será trabalhado qual aspecto da solução.

Uma característica importante de uma atividade[10] é o que ela expressa, ou seja, ela define **quem** participa da atividade, **quando** ela deve ser realizada, quais os **artefatos de entrada** e quais os **artefatos de saída**.

A Figura 4.3, mostra um exemplo de atividade. No canto superior esquerdo, encontra-se o(s) participante(s), ou seja, quem realiza, participa ou acompanha a atividade. No canto inferior esquerdo, estão as siglas dos artefatos de entrada, ou seja, os artefatos necessários para o bom desenvolvimento da atividade. No canto superior direito, estão os artefatos de saída, ou seja, artefatos criados ou atualizados na atividade. Finalizando, no centro, encontra-se a denominação dela.

Figura 4.3 - Exemplo de representação de uma atividade

4.5.1 Atividades da fase de solicitação

Este tópico explica a organização das atividades que compreendem a fase de solicitação, quais os seus objetivos e demais especificações necessárias.

4.5.1.1 Solicitar serviço

Esta atividade descreve como o cliente deve entrar em contato com a empresa de desenvolvimento de soluções de software para solicitar um novo serviço, ou como ele deve proceder em caso de solicitação de mudança de funcionalidade. Aqui o cliente irá contatar a empresa, em busca de solucionar suas necessidades, vindo a iniciar uma Solicitação.

[10] A forma de representação mostrada acima é denominada PEPP (AGUIAR e ROULIER, 2004) utilizada para representar o MA-MPS, é de propriedade SWQuality Consultoria e Sistemas e está distribuída sob a licença Creative Commons de "Atribuição - Uso Não Comercial 2.5 Brasil", podendo ser utilizada por terceiros desde que referenciada.

As solicitações geradas nesta atividade são de responsabilidade do Cliente, onde o artefato Solicitação representa o principal meio de comunicação ativo do Cliente e a empresa. Outros meios de comunicação podem ser adotados e especificados em um outro artefato denominado Plano de Comunicação, geralmente usado em grandes projetos.

4.5.1.2 Analisar necessidade

Essa atividade descreve como será realizada a atividade de análise de necessidades do Cliente que solicitou um serviço. Ela especifica também como observar o ambiente do Cliente em busca de conhecimentos mais detalhados sobre o problema em questão, e a forma como as pessoas que são afetadas por ele abordam a problemática e propõe soluções para os mesmos. Serão identificadas as necessidades expostas na Solicitação e então serão reunias a maior quantidade de informações possíveis sobre aquele contexto. Para essa atividade serão necessários o artefato de Visão de Negócio, Glossário e Regras de Negócio, sendo estes utilizados pelos papéis: Cliente e Analista de Requisitos.

O Analista de Requisitos será responsável por analisar o cenário da solicitação, as necessidades do Cliente, receber e propor sugestões de solução para o problema das partes envolvidas. Já o Cliente será responsável por disponibilizar essas informações e propor soluções para o problema.

4.5.1.3 Negociar proposta de solução

Esta atividade indica de que forma ocorrerá o processo de negociação da proposta de solução feita pelo Analista de Requisitos e Gerente de Projetos. Em visão geral, propõe-se uma reunião entre Cliente, Analista de Requisitos e Gerente de Projetos para que haja o esclarecimento da solução que será detalhada nas atividades consequentes. A proposta resulta na aceitação ou rejeição de um contrato firmado entre empresa e o Cliente para que sejam iniciadas as atividades de especificação da solução.

As propostas de solução serão discutidas observando a correspondência das necessidades no contexto especificado, sendo definida como solução a ser desenvolvida. Com base nesta definição, será elaborado o Contrato de Aceitação, o

qual será de responsabilidade do Gerente de Projetos. Nele serão definidos em detalhes do que se trata a funcionalidade em termos de preços e prazos.

Serão participantes desse processo: o (i) Analista de Requisitos, descrevendo as soluções propostas; (ii) o Gerente de Projetos, realizando as negociações e preenchendo a Ata de Reunião e; (iii) o Cliente, julgando, segundo seu entendimento, as soluções que serão especificadas, concordando ou não com os contratos.

4.5.2 Atividades da fase de análise

Este tópico exibe como serão realizadas as atividades de análise, identificação de requisitos e especificação dos mesmos. Para isso, serão expostas visões gerais, explicações e pontos importantes de cada uma das atividades.

4.5.2.1 Elicitar requisitos

Esta atividade descreve quais os passos necessários para que haja uma elicitação de requisitos de maneira concreta e assertiva. Ela é responsável pela definição funcional e não-funcional dos requisitos de software, incluindo a especificação dos diversos diagramas necessários para uma definição concreta das funcionalidades e características da solução de software. Para dar início a esta atividade, deve-se ter em mãos documentos básicos que descreva o ambiente e o problema que o Cliente enfrenta, sendo resultantes, documentos técnicos de modelagem da solução e protótipos para avaliação.

Como responsável por essa atividade temos o Analista de Requisitos. Ele reúne todas as informações inerentes ao projeto e elabora os artefatos: Documento de Elicitação de Requisitos, Especificação de Casos de Uso e Protótipos de Interface com o Usuário.

Esses documentos funcionarão como alicerce para o projeto, pois servirão como repositório principal de informações o no decorrer de sua execução do projeto.

4.5.2.2 Revisar requisitos

Esta atividade dá suporte à atividade de elicitação de requisitos através de uma revisão no conteúdo produzido objetivando eliminar ambiguidades, requisitos duplicados, erros de entendimento e identificar o relacionamento entre os requisitos. Nessa atividade o Cliente deve participar ativamente, identificando equívocos do Analista de Requisitos e sugerindo mudanças e definirão corretamente os requisitos. Os artefatos destinados à revisão são: Documento Elicitação de Requisitos e Documento de Especificação de Casos de Uso. Assim, o Analista de Requisito montará uma Matriz de Rastreabilidade tendo para isso o suporte do Cliente.

Essas atividades têm o objetivo principal de refinar as informações dos documentos, mantendo a fidelidade entre as informações nos documentos e o ambiente observado. Os documentos serão escritos de forma a simples e objetiva, onde todos os envolvidos no processo possam ter um entendimento fácil sobre os conceitos descritos no documento.

Nessa fase os documentos de Elicitação de Requisitos e Especificação de Casos de Uso são atualizados conforme conveniente ao projeto.

4.5.2.3 Definir tecnologia

Nessa atividade serão discutidas as questões referentes às tecnologias a serem empregadas no desenvolvimento do projeto. O analista de requisitos junto com o arquiteto de software irá definir como aproveitar melhor as tecnologias existentes observando o custo, acessibilidade e adaptabilidade à realidade do Cliente

Uma vez analisados todos os aspectos do contexto onde o problema esta inserido, o Arquiteto de Software reunirá informações de tecnologias que podem ser utilizadas para a resolução daquele problema.

Junto com o Analista de Requisitos, ele ira definir quais são as tecnologias que consiste no arcabouço conceitual que servirá de base para a construção do sistema.

4.5.2.4 Avaliar riscos e mensurar requisitos

Esta atividade explica como deve ser feita a atividade de avaliação de riscos de implementação e mensuração de requisitos. Aqui o Analista de Requisitos deve

identificar os riscos de implementação observando aspectos de tempo, possibilidade de desenvolvimento no cenário atual e viabilidade. Um ou mais desenvolvedores devem participar da mensuração, dando para todos os requisitos um número de pontos (horas de trabalho).

A ideia de mensurar os requisitos em pontos funciona da seguinte maneira. É apresentada a especificação de um requisito/caso de uso para um desenvolvedor, através de leitura ou apresentação gráfica. Em seguida, de acordo com o entendimento dele em relação ao desenvolvimento, serão atribuídas três notas: (i) a quantidade de pontos necessárias no pior caso de desenvolvimento do requisito, ou seja, quando o desenvolvedor não tem experiência e o problema pode ser mais complicado do que o que parece; (ii) a quantidade normal de pontos para ele desenvolver os requisitos e;.(iii) no melhor caso, a quantidade de pontos necessárias para ele desenvolver os requisitos, tendo em vista que tudo ocorrerá sem interferências e ele já tenha experiência em uma solução idêntica ou bem parecida com a apresentada.

Vale ressaltar que esse tipo de mensuração é apenas uma sugestão. Na prática, podem ser adotadas várias outras formas de mensurar os requisitos.

4.5.2.5 Definir diretrizes de teste

Aqui estão definidas as formas de realização dos testes sobre os requisitos da solução. Assim o Analista de Teste estima os resultados gerados pelo sistema como solução satisfatória para os problemas do Cliente. Dessa forma, um Plano de Testes é definido para dar suporte à realização fiel dos testes nas fases posteriores do processo de desenvolvimento. Nessa atividade o Analista de Testes irá analisar o Documento de Elicitação de Requisitos e definirá as diretrizes dos testes que serão aplicadas. Uma vez definidas as entradas, saídas e as condições de execução de cada teste, o Analista de Testes irá elaborar o Plano de Testes, que formalizará todo o procedimento de teste que será realizado sobre o sistema.

4.5.2.6 Negociar requisitos

Aqui estão definidas as atividades necessárias para a negociação de requisitos entre o Gerente de Projetos e o Cliente. Eles se reunirão para definir prioridades de

requisitos, estabelecer prazos e estipular metas. O Analista de Requisitos participa, esclarecendo e apresentando os requisitos elicitados e definidos. Finalizando com um contrato firmado entre o Cliente e a empresa de desenvolvimento.

4.5.3 Atividades da fase de planejamento

Este tópico mostra como são realizadas as atividades necessárias para a realização do planejamento para o desenvolvimento do projeto.

4.5.3.1 Planejar desenvolvimento

Nesta atividade estão descritos os passos necessários para que o planejamento do projeto seja realizado de forma satisfatória. O Gerente de Projetos, ao iniciar esta atividade, já deve possuir conhecimento sobre os requisitos, sobre o ambiente onde a solução será desenvolvida, testada e utilizada, sobre os recursos necessários para a execução do projeto, orçamento, prazos, sobre como se pretende desenvolver os requisitos etc. Assim ele tem embasamento suficiente para criar um bom Plano de Projeto, que seja satisfatório em prazo e viável em termos de custo.

O responsável por essa atividade é o Gerente de Projetos, que irá desenvolver o Plano de Projeto de Software. Esse documento irá reunir informações referentes recursos físicos e humanos a serem utilizados, no decorrer do desenvolvimento do projeto, bem como os prazos e metas a serem atingidas pela equipe.

4.5.3.2 Preparar ambiente de desenvolvimento

Esta atividade descreve como deve ser feita a preparação do ambiente de desenvolvimento de um software. A preparação, que será realizada pelo Gerente de Projetos, envolve tanto recursos humanos, como materiais. Os recursos humanos são: a alocação de funcionários para determinadas atividades e/ou a contratação de novos ou capacitação. Os recursos materiais são todos os equipamentos de hardware computacional e escritório, incluindo-se também as ferramentas de software, ou qualquer outro recurso necessário para o desenvolvimento adequado do processo.

4.5.4 Atividades da fase de desenvolvimento

Nesta fase são descritas todas as atividades necessárias para o desenvolvimento técnico de uma solução. Aqui são desenvolvidas todas as especificações arquiteturais e codificação de uma solução.

4.5.4.1 Definir arquitetura

Atividade feita pelo Arquiteto de Software, sendo uma das mais importantes para o sucesso do projeto. Aqui serão definidas todas as especificações técnicas (diagramas de classe UML, por exemplo) da solução na tecnologia definida na fase de análise. Enfim, todo o arcabouço necessário para a codificação consistente da solução. O Arquiteto de Software ira desmembrar o documento de o documento de Arquitetura de Software e irá detalhá-lo de forma que possa ser executado pela equipe de desenvolvimento. Nessa fase será elaborado o plano de Integração do Build, que terão formatados todos os conceitos técnicos que irão estruturar o sistema.

4.5.4.2 Codificar

A atividade de codificação do software define os passos necessários para criar todo o código fonte de todos os métodos predefinidos na atividade de arquitetar os requisitos. Aqui o desenvolvedor cria, revisa e comenta o código fonte de forma eficiente e facilite a leitura posterior. Segundo definido nas fases anteriores, serão construídos os algoritmos que constituirão o produto de software, que obedecerão ao padrão de construção definido no Modelo de Implementação do Software.

4.5.5 Atividades da fase de teste

Nesta fase são definidos todos os testes necessários para a avaliação das funcionalidades desenvolvidas. Nesse passo, são usadas como referência as diretrizes de testes desenvolvidas na fase de análise.

4.5.5.1 Testar

A atividade de testar o software, embora simbolizada como única, envolve vários tipos de testes (descritos na seção Fluxo Principal). Os testes verificam vários

aspectos do funcionamento do software. Os resultados destes testes são verificados e remetidos para as atividades mais convenientes, para serem corrigidos quando resultam em erro.

- **Teste de integridades dos dados** – Esse teste verifica, por exemplo, se a precisão dos cálculos está dentro do especificado ou se os dados são armazenados e exibidos ao usuário de forma correta e concisa.
- **Teste de interface com o usuário** – verifica se todos os controles (botões, caixas de listagem, botões de checagem, etc.) estão funcionando perfeitamente. Também são verificadas as posições das caixas de mensagem exibidas e observação da formação dos controles durante o dimensionamento das janelas.
- **Teste de segurança** – verifica-se a segurança das informações que trafegam entre o usuário e o sistema (WEB ou LAN) ou se dados estão sendo exibidos para papéis não autorizados
- **Teste de funcionamento** – verifica através da simulação de um caso de uso real, se o funcionamento do sistema está dentro do esperado.
- **Teste de aceitação** – esse teste observa a aceitação do cliente em relação à(s) funcionalidade(s) desenvolvida(s). É esperado que durante esse teste seja gerada alguma solicitação de mudança, já que esse é o primeiro contato do cliente com a solução.
- **Teste de integração** – um dos testes mais importantes para a escalabilidade do sistema. Esse teste observa todos os aspectos da integração da nova funcionalidade com o sistema (em caso de ser uma integração de uma nova funcionalidade a um sistema existente).

Durante a realização dos testes, normalmente são encontradas falhas, inconsistências e irregularidades, dependendo de qual seja, a correção é encaminhada para a fase e atividade responsável por resolvê-las.

Uma funcionalidade só passará para a fase seguinte se ela passar em todos os testes submetidos. Isso especifica que ela está em estado de aceitação e funcionamento.

4.5.6 Atividades da fase de entrega

Esta fase compreende todas as atividades necessárias para a entrega da(s) solução(ões). Nesta fase, inicialmente são definidas como serão realizadas as implantações e treinamentos aos usuários, necessitando de uma atenção maior dos responsáveis em busca de melhorar cada vez mais o relacionamento com o cliente. Em seguida serão executados todos os treinamentos e a solução finalmente será implantada.

4.5.6.1 Planejar treinamento e integração

Nessa atividade deve ser feito o planejamento para a realização dos treinamentos dos usuários finais, além do planejamento de como será feita a implantação da solução no ambiente do cliente. Planejamentos que levarão à criação dos artefatos Plano de Treinamento e Plano de Implantação. Estes serão criados através de um ou mais acordos firmados entre o cliente e a empresa de desenvolvimento, devidamente documentados na Ata de Reunião, acordos visando suprir as necessidades do cliente em termo de tempo, espaço e disponibilidade no ambiente dele.

4.5.6.2 Treinar usuários

Esta atividade descreve como devem ser feitos os treinamentos dos usuários finais para que eles estejam preparados para a implantação e utilização da nova solução de software. Treinar os usuários da solução é uma tarefa muito importante e deve ser feita antes da implantação.

Sabe-se que o ser humano naturalmente possui resistência à mudança e por isso é importante treinar os usuários de uma solução antes da utilização efetiva. Isso é necessário porque, geralmente, é adicionada alguma mudança nos procedimentos normais de uma atividade que pode passa a ser totalmente ou parcialmente automatizada por um software, visando a exclusão ou diminuição de entraves típicos das atividades manuais. Assim os treinamentos devem apresentar a nova solução esclarecer as dúvidas dos usuários até que eles sintam-se seguros.

4.5.6.3 Implantar solução e dar suporte

Esta atividade descreve como será feita a implantação da solução previamente planejada e também do suporte ao cliente.

No caso da implantação, é importante ter um bom conhecimento sobre o ambiente do cliente, por exemplo: se o cliente for um supermercado que deseja automatizar as operações no caixa. É importante que a implantação seja executada fora do horário de atendimento para não atrapalhar o atendimento ao cliente, ou possa vir a gerar uma situação que venha a causar transtornos ao cliente.

No caso do suporte, as opções disponibilizadas pela empresa serão oferecidas ao cliente e incluídas no contrato antes do início das atividades de especificação da solução. Assim, o cliente poderá ter várias opções de suporte disponíveis de acordo com os seus interesses. Alguns exemplos são:

- Suporte via telefone;
- Suporte online (mensagem instantânea, e-mail, etc.);
- Suporte presencial (envio de um profissional até o cliente);
- Manual de usuário impresso ou em mídia eletrônica;
- Assistência remota.

As possibilidades de suporte ao usuário são inúmeras. Nesse momento uma empresa deve oferecer o melhor serviço possível, de forma a minimizar os esforços do cliente na busca de esclarecimentos sobre dúvidas.

4.6 PROCESSOS DE APOIO

4.6.1 Gerenciamento das Mudanças (GMD)

A fase de mudança corresponde as atividades referentes à alteração, exclusão ou inserção de requisitos de projeto durante o desenvolvimento. Podemos dizer que as atividades que compreendem essa fase estão relacionadas a eventos que façam com que o processo tome uma caminho diferente do habitual (normal). Como a desistência de uma funcionalidade, modificação em outra, desistência do projeto, etc.

4.6.1.1 Analisar solicitação

Nessa atividade serão analisadas as influências que a mudança solicitada irá acarretar ao projeto. Será analisado se algum requisito será invalidado ou adicionado no projeto por uma solicitação do cliente ou ainda se o processo terá que parar o desenvolvimento de alguma funcionalidade. Uma vez mensurados os impactos da solicitação de mudança, serão avaliados os gastos referentes a mudança segundo as horas trabalhadas. Essa atividade será executada pelo Gerente de Projetos com ajuda do Analista de requisitos e do cliente. Como resultado dessa atividade será atualizado o documento de Elicitação de Requisitos e a matriz de rastreabilidade.

4.6.1.2 Negociar solução com o cliente

Nesta atividade serão apresentados ao cliente o impacto que as mudanças solicitadas irão acarretar ao projeto e os gastos provenientes da mesma. Nessa reunião, o Gerente de Projetos junto ao Analista de Requisitos, irão negociar a viabilidade das alterações com o cliente sugerindo alternativas e soluções. Então serão definidas as inserções, mudanças ou cancelamentos de requisitos ou do projeto como um todo. Essas atividades serão executadas pelo Gerente de Projetos, junto com o Analista de Requisitos e o Cliente. As definições acertadas nessa fase serão organizadas no contrato de aceitação que terá a assinatura de todos os papeis envolvidos na atividade e servirá para validar a necessidade do cliente.

4.6.1.3 Executar mudança planejada

Nessa fase serão executadas as mudanças definidas na reunião com o Cliente. Caso seja acordado o cancelamento do projeto, serão apresentados todos os encargos ao Cliente segundo estabelecido no contrato.

Caso a solicitação consista em numa mudança em uma funcionalidade da solução de software, o Gerente de Projeto e o Analista de Requisitos atualizarão os artefatos para a realização das alterações. Durante esse processo, alguns requisitos podem vir a ser invalidados por vários motivos. Entre eles podemos citar: (i) mudanças na regras de negócio do Cliente; (ii) não há mais a necessidade. Para saber exatamente quais requisitos, características e recursos do processo e do produto

serão afetados pela mudança, deverá ser usada uma Matriz de Rastreabilidade que indicará o relacionamento entre os vários componentes do produto de software. Nesse caso, “componentes” pode ser entendido como características da solução ou artefatos. Os requisitos que foram invalidados segundo a Matriz de Rastreabilidade serão excluídos e/ou substituídos e acoplados ao projeto, e então será gerada uma nova solicitação comum e o processo seguirá normalmente.

4.6.2 Verificação da qualidade (VQA)

Nessa fase serão observados os fatores determinantes da qualidade da produção realizada naquele momento. As atividades de validação serão executadas na passagem de fase do processo e analisarão os documentos produzidos verificando se obedecem aos padrões da empresa e se possuem algum tipo de inconformidade. Essa fase objetiva aumentar a qualidade dos documentos relacionados ao sistema e consequentemente sua própria.

4.6.2.1 Levantamento do projeto

A atividade onde é feita a análise do andamento do projeto, se o mesmo está sendo realizado conforme o planejado e se obedece aos padrões de qualidade definidas para o processo e para o produto. Também é feita uma análise dos problemas e dificuldades com o fim de evitar que estes venham a ocorrer. Essa atividade é desenvolvida pelo Analista de Requisitos que observará os problemas que atrapalharam a fase e os principais efeitos causados no processo através da comparação das atividades realizadas e as programadas no plano de projeto de software.

4.6.2.2 Avaliação da qualidade

O Analista de Qualidade irá revisar as evidências para certificar que o processo de os padrões para qualidade do produto estejam sendo observados. O objetivo dessa atividade é manter a qualidade na construção do produto e na execução das atividades conforme observado no processo, o que facilitará futuras manutenções e fácil entendimento dos envolvidos no processo.

4.6.2.3 Reunião

O Analista de Qualidade irá passar sua avaliação acerca da análise realizada ao gerente, para que este possa ter um maior controle sobre o andamento do projeto, já que pode haver a necessidade de realizar algum realinhamento ao que foi planejado, como também manter a aderência da execução dos procedimentos aos padrões de qualidade institucionalizados.

4.6.2.4 Correção

O Analista de Qualidade irá comunicar os desvios aos padrões aos responsáveis. Os artefatos que possuírem inconformidades serão devolvidos aos seus respectivos responsáveis para que eles façam as devidas alterações. Uma vez as alterações realizadas, o processo seguirá seu curso normal.

5 CONSIDERAÇÕES FINAIS

Tendo em vista todas as informações reunidas para o desenvolvimento desse projeto, observa-se que muito há de fazer e estudar sobre a melhoria do desenvolvimento de software, principalmente quanto ao desenvolvimento de software por pequenas empresas. O objeto alcançado neste projeto foi a modelagem de um processo de desenvolvimento reunindo características que o torne simples, porém, completo o suficiente para que, equipes e empresas iniciantes ou não, tenham todo o arcabouço para seguirem durante o desenvolvimento dos seus projetos. Foi observado também que o sucesso na implantação de um processo de software está intimamente ligado às características do problema a ser atacado e aos recursos disponíveis para sua resolução.

O processo aqui apresentado e descrito está apto a gerir as problemáticas envolvidas no desenvolvimento de um software, de forma que, qualquer profissional que tenha o mínimo de conhecimento dos conceitos da Engenharia de Software possa desenvolvê-lo sem dificuldades atingindo um alto grau de qualidade.

Em resumo, no decorrer do desenvolvimento do projeto foi reunida uma carga de conhecimentos que irá contribuir para a vida profissional dos participantes da equipe envolvida no desenvolvimento do Prumo. Além de resultar em um processo de software que permitirá a profissionais iniciantes, desenvolverem um projeto seguindo os conceitos de alguns dos principais autores contemporâneos sobre o tema deste trabalho. Adicionalmente, seguindo as melhores práticas dos processos, metodologias e *frameworks* mais utilizados no mercado. Portanto engenharia é qualidade.

REFERÊNCIAS

AGUIAR, H. V.; ROULIER, A. C. Primitivas para Definição de Processo - PEPP. **SWQuality**, Recife, 2004. Disponivel em: <www.swquality.com.br/pepp/>. Acesso em: 5 Dezembro 2009.

AMBER, S. W. An Introduction to Processo Patterns. **Site da Amby Soft**, 1998. Disponivel em: <http://www.ambysoft.com/downloads/ProcessPatterns.pdf>. Acesso em: 4 de Novembro de 2009.

AMBER, S. W. **Process Patterns:** Building Large-Scale Systems Using Technology. Cambridge: Cambridge University Press, 1998.

BONNER, W. Setor de tecnologia da informação cresce com a crise. **Site da Globo.com/jornalnacional**, São Paulo, 13 de Fevereiro de 2009. Disponivel em: <http://video.globo.com/Videos/Player/Noticias/0,GIM965786-7823-SETOR+DE+TECNOLOGIA+DA+INFORMACAO+CRESCE+COM+A+CRISE,00.html>. Acesso em: 26 de Outubro de 2009.

DESENVOLVIMENTO Iterativo e Incremental. **Wikipédia, a enciclopédia livre**, 17 Outubro 2009. Disponivel em: <http://pt.wikipedia.org/wiki/Desenvolvimento_iterativo_e_incremental>. Acesso em: 26 de Outubro de 2009.

DUTRA, L. R. Paradigmas de Engenharia de Software. **Site da Universidade de Brasília**, 2008. Disponivel em: <http://www.redes.unb.br/material/Metodologia%20de%20Desenvolvimento%20de%20Software/aula3.pdf>. Acesso em: 8 de Novembro de 2009.

FILHO, E. M. G. et al. Tributação e Desenvolvimento no Setor de Software Brasileiro. **Site da Abes - Associação Brasileira das Empresas de Software**, Dezembro 2006. Disponivel em: <http://www.faltaesselink.com.br>. Acesso em: 15 de Outubro de 2009.

INTERNATIONAL BUSINESS MACHINES - IBM. IBM Rational Unified Process. **IBM - International Business Machines**, 2009. Disponivel em: <http://www-01.ibm.com/software/awdtools/rup/>. Acesso em: 14 de Novembro de 2009.

ISO/IEC 12207. 12207:2008(E) Systems and software engineering - Software life cycle processes. **Site da ISO - International Organization for Standardization**, 2008. Disponivel em: <http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=43447>. Acesso em: 4 de Novembro de 2009.

MACHADO, C. Â. F.; WEBER, K. C. **Qualidade e Produtividade em Software**. São Paulo: Makron Books, 2001.

MELO, C. D. O. Reutilização de Software: Classificação e Seleção de Artefatos Reutilizáveis. **Site do Instituto de Matemática e Estatística da USP**, São Paulo, 25 de Junho de 2004. Disponivel em: <http://www.ime.usp.br/~yw/2004/mac5701i/monografias/claudia-acvm.pdf>. Acesso em: 28 de Outubro de 2009.

MPS.BR. Implementação do MPS.BR em organizações do tipo fábica de software. **Guia de Implementação - Parte 9**, 2009, 14 de Novembro de 2009. 4-6.

PRESSMAN, R. S. **O papel evolutivo do software**. Tradução de Carlos Barbosa Santos. São Paulo: Makron Books do Brasil Editor Ltda, 1995. 4-6 p. ISBN ISBN: 8586804576.

PRESSMAN, R. S. **Engenharia de Software**. 6. ed. [S.l.]: McGraw-Hill, 2006. 17-55 p. ISBN 8586804576.

PROCESSO de Desenvolvimento de Software. **Wikipédia, a enciclopédia livre**, 17 de Outubro de 2009. Disponivel em: <http://pt.wikipedia.org/wiki/Processo_de_desenvolvimento_de_software>. Acesso em: 26 de Outubro de 2009.

RATIONAL SOFTWARE CORPORATION. Artefato: Ator. **Site da Wthreex**, 2002. Disponivel em: <http://www.wthreex.com/rup/process/artifact/ar_actor.htm>. Acesso em: 5 de Novembro de 2009.

RATIONAL SOFTWARE CORPORATION. Papel: Arquiteto de Software. **Site da WThreex**, 2002. Disponivel em: <http://www.wthreex.com/rup/portugues/index.htm>. Acesso em: 31 de Outubro de 2009.

SEBRAE. Critérios e conceitos para classificação de empresas. **Site do SEBRAE**, João Pessoa, 19 de Maio de 2005. Disponivel em: <http://www.sebrae.com.br/customizado/estudos-e-pesquisas/integra_bia?ident_unico=97>. Acesso em: 06 de Outubro de 2009.

SIMÃO, I.; VARELA, P. A Engenharia de Requisitos como processo inovador nas organizações. **Site da RUN - Repositório Universidade Nova**, Monte de Caparica, Julho 2009. Disponivel em: <http://dspace.fct.unl.pt/bitstream/10362/1972/1/WPSeries_08_2009ISimao_PVarelaB.pdf>. Acesso em: 26 de Outubro de 2009.

SOFTEX. MPS.BR, capacitação e empreendedorismo. **Guia de Implementação – Parte 9:** Implementação do MR-MPS em organizações do tipo Fábrica de Software, 02 Outubro 2009. ISSN ISBN 978-85-99334-16-4. Disponivel em: <http://www.softex.br/mpsbr/_guias/guias/MPS.BR_Guia_de_Implementacao_Parte%209.pdf>. Acesso em: 12 de Outubro de 2009.

SOMMERVILLE, I. **Engenharia de Software**. 6. ed. [S.l.]: Addison Wesley, 2001. 42-54 p.

UENO, M. www.assespro.org.br. **Site da ASSESPRO - Associação das Empresas de Tecnologia da Informação, Software e Internet**, Abril 2007. Disponivel em: <http://www.assespro.org.br/images/pdti.pdf>. Acesso em: 12 de Outubro de 2009.

WIKIPÉDIA, A ENCICLOPÉDIA LIVRE. IBM Rational Unified Process. **Site Wikipédia, a enciclopédia livre**, 2009. Disponivel em: <http://pt.wikipedia.org/wiki/IBM_Rational_Unified_Process>. Acesso em: 2009 de Novembro de 14.

WIKIPÉDIA, A ENCICLOPÉDIA LIVRE. Modelo Cascata. **Wikipédia, A enciclopédia livre**, 17 de Outubro de 2009. Disponivel em: <http://pt.wikipedia.org/wiki/Modelo_em_cascata>. Acesso em: 26 de Outubro de 2009.

APÊNDICE A - CICLO DE VIDA E MAPA DE ATIVIDADES

Mapa do processo

Mapa dos processos de apoio

APÊNDICE B – PAPÉIS

APÊNDICE B.1 – CLIENTE

Processo de desenvolvimento de software

Detalhamento do Papel

Cliente

Descrição

Este papel representa a pessoa que entende a regra de negócio da empresa a qual se destina a solução. Ele pode ser representado por uma ou mais pessoas, dependendo do projeto e da necessidade. Dois perfis distintos podem ser analisadas, a do cliente externo: aquele enfrenta um problema em um ambiente qualquer e não trabalha na empresa que desenvolve a solução. O outro seria o cliente interno: aquele que trabalha na empresa que desenvolve a solução e conhece a regra de negócio da empresa a qual se destina a solução.

Atribuições

- ✓ Solicitar serviços.
- ✓ Solicitar mudanças de software.
- ✓ Assinar contratos de proposta e aceitação.
- ✓ Descrever seus problemas de forma clara e entendível.
- ✓ Especificar a Regras de Negócio.
- ✓ Colaborar para o desenvolvimento de um Glossário.

Artefatos do papel

- ✓ Documentos trazidos por ele para ajudar no entendimento do problema.
- ✓ Glossário.
- ✓ Regras de Negócio.
- ✓ Solicitação

Títulos

Exigidos

- ✓ Nenhum.

Desejados

- ✓ Ensino médio completo.
- ✓ Conhecimentos básicos em informática.
- ✓ Qualquer curso, com duração mínima de 20 horas e que desenvolva contato com cliente.

Características

- ✓ Receptivo.
- ✓ Comunicativo.
- ✓ Observador.
- ✓ Clareza de idéias.

APÊNDICE B.2 – ANALISTA DE REQUISITOS

Processo de desenvolvimento de software

Detalhamento do Papel

Analista de Requisitos

Descrição

Este papel representa a pessoa que sabe identificar as necessidades do cliente e propor soluções para elas. Ele detalha a especificação de uma parte da funcionalidade do sistema, descrevendo os requisitos de um ou mais casos de uso, criando e detalhando-os.

Atribuições

- ✓ Analisar necessidades do cliente.
- ✓ Propor soluções para as necessidades do cliente.
- ✓ Apresentar as soluções propostas.
- ✓ Elícitar e detalhar requisitos.
- ✓ Criar e detalhar casos de uso.
- ✓ Criar matrizes de rastreabilidade.
- ✓ Avaliar riscos na implementação de requisitos.

Artefatos do papel

- ✓ Visão de Negócio.
- ✓ Documento de Elicitação de Requisitos.
- ✓ Documento de Casos de Uso.
- ✓ Matrizes de Rastreabilidade.

Títulos

Exigidos

- ✓ Curso superior na área de tecnologia da informação.
- ✓ Um ano de experiência como analista de requisitos.

Desejados

- ✓ Especialização em engenharia de software.
- ✓ Ter participado de pelo menos cinco projetos de software.

Características

- ✓ Habilidade para identificar problemas.
- ✓ Boa comunicação interpessoal.
- ✓ Comunicativo.
- ✓ Receptivo.

APÊNDICE B.3 – GERENTE DE PROJETOS

Processo de desenvolvimento de software

Detalhamento do Papel

Gerente de Projetos

Descrição

Representa um líder. Uma pessoa habilitada para assumir este papel deve ter habilidade para liderar equipes de projeto. Acompanha todas atividades do processo e realiza as negociações com o cliente. Um gerente de projeto não necessariamente deve ter conhecimentos técnicos para que possa exercer eficientemente suas atividades dentro da empresa.

Atribuições

- ✓ Liderar a equipe de desenvolvimento de soluções.
- ✓ Adequar a equipe ao projeto.
- ✓ Criar Planos de Projeto.
- ✓ Realizar negociações junto ao cliente.
- ✓ Preparar ambiente da empresa para acomodar os projetos.
- ✓ Providenciar recursos materiais (Hardware e Software).

Artefatos do papel

- ✓ Contratos.
- ✓ Planos de Projeto de Software.
- ✓ Documento de Riscos.
- ✓ ATA.
- ✓ Gestão de Ações Corretivas.

Títulos

Exigidos

- ✓ Curso superior em tecnologia da informação ou administração.
- ✓ Experiência de um ano como Líder de equipe ou gerente de projetos.

Desejados

- ✓ Ter conhecimento em gerenciamento de projetos com PMBOK e/ou CMMI.
- ✓ Ter especialização em gerência de projetos de software.

Características

- ✓ Habilidade para liderar equipes.
- ✓ Boa comunicação interpessoal.
- ✓ Comunicativo.
- ✓ Receptivo.
- ✓ Saber ouvir os integrantes da equipe.
- ✓ Capacidade de estimular o trabalho em equipe.

APÊNDICE B.4 – ARQUITETO DE SOFTWARE

Processo de desenvolvimento de software

Detalhamento do Papel

Arquiteto de software

Descrição

Este papel representa a pessoa capaz de definir a arquitetura de uma solução de software. Ele deve ter conhecimento sobre padrões de projeto e implementação de software, um exemplo seria UML, além de conhecer várias tecnologias e frameworks. O arquiteto de software conhece ferramentas que auxiliam no desenvolvimento de um bom software.

Atribuições

- ✓ Desenvolver diagramas de atividade.
- ✓ Desenvolver diagramas de classe.
- ✓ Desenvolver diagramas entidade-relacionamento.
- ✓ Desenvolver tabelas de bancos de dados.
- ✓ Desenvolver plano de integração (se necessário) identificando subsistemas e analisando os pontos de integração.

Artefatos do papel

- ✓ Plano de Arquitetura de Software.
- ✓ Modelo de Implementação.
- ✓ Plano de Integração.

Títulos

Exigidos

- ✓ Curso superior em tecnologia da informação.
- ✓ Experiência de um ano como analista de sistemas ou arquiteto de software.

Desejados

- ✓ Especialização em análise e desenvolvimento de sistemas.
- ✓ Certificações e/ou comprovação de conhecimento em análise e desenvolvimento de sistemas.

Características

- ✓ Competência.
- ✓ Boa comunicação interpessoal.
- ✓ Capacidade para trabalhar e/ou liderar equipes.
- ✓ Capacidade de desenvolver boas soluções.

APÊNDICE B.5 – DESENVOLVEDOR

Processo de desenvolvimento de software

Detalhamento do Papel

Desenvolvedor

Descrição

Este papel representa a pessoa capaz de traduzir problemas do "mundo humano" para o "mundo computacional" através de linguagens. O desenvolvedor deve conhecer os princípios básicos da computação, como algoritmos e estruturas de dados, além de saber aplicá-los em uma ou mais linguagens de programação. O desenvolvedor dever saber trabalhar em equipe dividir o conhecimento com os demais colegas de trabalho.

Atribuições

- ✓ Codificar e documentar soluções de software.
- ✓ Criar mecanismos para uma melhor implementação das soluções.
- ✓ Utilizar os padrões de codificação estabelecidos pela empresa.
- ✓ Mensurar requisitos de software.
- ✓ Avaliar riscos de codificação.
- ✓ Fornecer feedback das atividades para o líder da equipe.

Artefatos do papel

- ✓ Plano de integração.
- ✓ Componente de Software.

Títulos

Exigidos

- ✓ Curso técnico na área de desenvolvimento de sistemas.

Desejados

- ✓ Curso superior na área de TI com foco em desenvolvimento de software.

Características

- ✓ Competência.
- ✓ Agilidade para codificar soluções.
- ✓ Facilidade de perceber de problemas.
- ✓ Saber trabalhar em equipe.

APÊNDICE B.6 – ANALISTA DE TESTE

Processo de desenvolvimento de software

Detalhamento do Papel

Analista de testes

Descrição

Este papel representa a pessoa habilitada para desenvolver testes de software. Ele deve atuar na fase de Análise, onde cria as propostas de testes esperadas para as funcionalidades elicitadas, e principalmente, na fase de teste de software, elaborando o plano de teste de software e solicitando mudanças.Ele executará os testes definidos no plano de testes e agrupara os resultados no plano de testes , onde serão analisados e processados conforme descrito no plano de testes.

Atribuições

- ✓ Definir diretrizes de teste.
- ✓ Elaborar plano de teste.
- ✓ Criar solicitações de mudanças.
- ✓ Analisar relatórios de teste.
- ✓ Realizar testes.

Artefatos do papel

- ✓ Plano de teste.
- ✓ Solicitações de mudança.

Títulos

Exigidos

- ✓ Curso superior em tecnologia da informação.
- ✓ Experiência de um ano como analista de teste.

Desejados

- ✓ Especialização em teste de software.
- ✓ Especialização em qualidade de software.

Características

- ✓ Saber trabalhar em equipe
- ✓ Competência
- ✓ Boa comunicação interpessoal
- ✓ Facilidade para identificar soluções
- ✓ Capacidade para sugerir boas soluções

APÊNDICE B.7 – ANALISTA DE SUPORTE

Processo de desenvolvimento de software

Detalhamento do Papel

Analista de Suporte

Descrição

O papel de analista de suporte deve ser atribuído a uma pessoa capaz de realizar treinamento sobre sistemas computacionais ou empresariais. Ele deve ser comunicativos e ter boa relação com clientes. O analista de suporte desenvolve e realiza todos os treinamentos necessários para a utilização correta da solução de software.

Atribuições

- ✓ Comunicar-se com o cliente.
- ✓ Desenvolver e executar treinamento de usuários.
- ✓ Desenvolver e executar implantação da solução.
- ✓ Dar suporte ao usuário.

Artefatos do papel

- ✓ Plano de Treinamento.
- ✓ Plano de Implantação.
- ✓ Solicitação de Treinamento.
- ✓ Comprovante de Implantação da Funcionalidade.

Títulos

Exigidos

- ✓ Curso superior em tecnologia da informação
- ✓ Experiência em ministrar aulas e/ou cursos

Desejados

- ✓ Cursos que envolvam instalação e manutenção de aplicativos e sistemas operacionais.
- ✓ Cursos que envolvam instalação e manutenção de servidores.

Características

- ✓ Bom relacionamento interpessoal.
- ✓ Comunicativo.
- ✓ Saber ouvir as opiniões do cliente.
- ✓ Clareza de idéias.

APÊNDICE B.8 – ANALISTA DE QUALIDADE

Processo de desenvolvimento de software

Detalhamento do Papel

Analista de qualidade

Descrição

Este papel é representado pela pessoa que realizará as auditorias sobre os artefatos produzidos durante o ciclo de vida do processo. Ele observa os artefatos, examinando-os segundo os padrões estabelecidos pela empresa de desenvolvimento.

Atribuições

- ✓ Realizar auditorias no decorrer do processo.
- ✓ Zelar pelo alto nível de qualidade do software.
- ✓ Verificar a conformidade das atividade realizadas.
- ✓ Verificar a conformidade dos documentos criados e atualizados no decorrer do processo.

Artefatos do papel

- ✓ Lista de verificação de qualidade.
- ✓ Todos os documentos das fases anteriores.

Títulos

Exigidos

- ✓ Curso Superior em Tecnologia da Informação
- ✓ Experiência de um Ano

Desejados

- ✓ Especialização em Qualidade de Software
- ✓ Especialização em Engenharia de Software

Características

- ✓ Comunicativo
- ✓ Observador
- ✓ Bom Relacionamento Interpessoal
- ✓ Saber Ouvir as Opiniões da Equipe

APÊNDICE C – ARTEFATOS

APÊNDICE C.1 – SOLICITAÇÃO

SOLICITAÇÃO

<Nome do projeto>

Solicitante: <Nome do solicitante>
Papel/função: <Nome do papel ou função>

Perfil da solicitação: ( ) Novo serviço ( ) Mudança ( )Treinamento

Descrição da mudança solicitada:_____

_____

_____

_____

_____

Impactos previstos:

No cronograma	*<Quantidade adicionada ou reduzida de dias>*
No custo	*<Valor do aumento ou redução no previsto>*
Na qualidade	*<Possível(is) impacto(s) na qualidade>*
Nos riscos	*<Novo(s) risco(s) observado(s)>*

Parecer do responsável: ( ) Aprovado ( )Reavaliar/Renegociar

Observações:_____

_____

_____

_____

12 de março de 2010, _____

Assinatura do solicitante

12 de março de 2010, _____

Assinatura do responsável

APÊNDICE C.2 – DOCUMENTO DE VISÃO

<Nome da Empresa>
<Slogan>

<Rua>, Nº <Número> - <Bairro> - <Cidade> - <UF>
CEP: <CEP>

<NOME DO PROJETO>

DOCUMENTO DE VISÃO

VERSÃO 1

Histórico de revisão

Data	Versão	Descrição	Autor

SUMÁRIO

1 VISÃO 117
1.1 INTRODUÇÃO 117
1.2 FINALIDADE 117
1.3 ESCOPO 117
1.4 DEFINIÇÕES, ACRÔNIMOS E ABREVIAÇÕES 117
1.5 REFERÊNCIAS 117
1.6 VISÃO GERAL 118
2 POSICIONAMENTO 119
2.1 OPORTUNIDADE DE NEGÓCIOS 119
2.2 DESCRIÇÃO DO PROBLEMA 119
2.3 SENTENÇA DE POSIÇÃO DO PRODUTO 119
3 DESCRIÇÕES DOS ENVOLVIDOS E USUÁRIOS 121
3.1 DEMOGRAFIA DO MERCADO 121
3.2 RESUMO DOS ENVOLVIDOS 121
3.3 RESUMO DOS USUÁRIOS 122
3.4 AMBIENTE DO USUÁRIO 122
3.5 PERFIS DOS ENVOLVIDOS 123
3.6 <NOME DO ENVOLVIDO> 123
3.7 PERFIS DE USUÁRIOS 123
3.7.1 <NOME DO USUÁRIO> 124
3.8 RINCIPAIS NECESSIDADES DOS USUÁRIOS OU DOS ENVOLVIDOS 124
3.9 ALTERNATIVAS E CONCORRÊNCIA 125
3.9.1 <UMCOMPETIDOR> 125
3.9.2 <OUTROCOMPETIDOR> 125
4 VISÃO GERAL DO PRODUTO 126
4.1 PERSPECTIVA DO PRODUTO 126
4.2 RESUMO DOS RECURSOS 126
4.3 SUPOSIÇÕES E DEPENDÊNCIAS 126
5 RECURSOS DO PRODUTO 127
5.1 <UM RECURSO> 127
5.2 <OUTRO RECURSO> 127
6 RESTRIÇÕES 128

7 FAIXAS DE QUALIDADE 129
8 PRECEDÊNCIA E PRIORIDADE 130
9 OUTROS REQUISITOS DO PRODUTO 131
9.1 PADRÕES APLICÁVEIS 131
9.2 REQUISITOS DO SISTEMA 131
9.3 REQUISITOS DE DESEMPENHO 131
9.4 REQUISITOS AMBIENTAIS 131
10 REQUISITOS DA DOCUMENTAÇÃO 132
10.1 MANUAL DO USUÁRIO 132
10.2 AJUDA ON-LINE 132
10.3 GUIAS DE INSTALAÇÃO E DE CONFIGURAÇÃO, E ARQUIVO LEIAME 132
10.4 ROTULAÇÃO E EMBALAGEM 132
11 ATRIBUTOS DE RECURSOS 133
11.1 STATUS 133
11.2 BENEFÍCIO 133
11.3 ESFORÇO 133
11.4 RISCO 134
11.5 RELEASE-ALVO 134

1 VISÃO

1.1 INTRODUÇÃO

[A finalidade deste documento é coletar, analisar e definir as necessidades e características de nível superior do <<Nome do Sistema>>. Ele se concentra nos recursos necessários aos envolvidos e aos usuários-alvo e nas razões que levam a essas necessidades. Os detalhes de como o <<Nome do Sistema>> atende a essas necessidades estão descritos nas especificações suplementares e de caso de uso.]

[A introdução do documento de Visão oferece uma visão geral de todo o documento. Ela contém a finalidade, o escopo, as definições, os acrônimos, as abreviações, as referências e a visão geral deste documento Visão.]

1.2 FINALIDADE

[Especifique a finalidade deste documento Visão.]

1.3 ESCOPO

[Uma breve descrição do escopo deste documento Visão; a que Projeto(s) ele está associado e tudo o mais que seja afetado ou influenciado por este documento.]

1.4 DEFINIÇÕES, ACRÔNIMOS E ABREVIAÇÕES

[Esta subseção fornece as definições de todos os termos, acrônimos e abreviações necessárias à adequada interpretação do documento Visão. Essas informações podem ser fornecidas mediante referência ao Glossário do projeto.]

1.5 REFERÊNCIAS

[Esta subseção apresenta uma lista completa de todos os documentos mencionados no documento de Visão. Identifique cada documento por título, número do relatório (se aplicável), data e organização de publicação. Especifique as fontes a partir das quais as referências podem ser obtidas. Essas informações podem ser fornecidas por um anexo ou outro documento.]

1.6 VISÃO GERAL

[Esta subseção descreve o que o restante do documento Visão contém e explica como o documento está organizado.]

2 POSICIONAMENTO

2.1 OPORTUNIDADE DE NEGÓCIOS

[Faça uma breve descrição da oportunidade de negócios atendida por este projeto.]

2.2 DESCRIÇÃO DO PROBLEMA

[Forneça uma descrição resumindo o problema que está sendo resolvido pelo projeto. Pode ser utilizado o seguinte formato:]

O problema é	
Que afeta	
Cujo impacto é	
Uma boa solução seria	

2.3 SENTENÇA DE POSIÇÃO DO PRODUTO

[Forneça uma sentença geral resumindo, no nível mais alto, a posição exclusiva que o produto pretende ocupar no mercado. Pode ser utilizado o seguinte formato:]

Para	*[cliente-alvo]*
Que	*[indique a necessidade ou oportunidade]*
O <Nome do produto> é um (a)	*[categoria do produto]*
Que	*[indique o principal benefício, ou seja, o motivo que leva a comprar]*
Diferente de	*[principal alternativa da concorrência]*

Nosso produto	*[indique a principal diferença]*

[Uma sentença de posição do produto comunica o objetivo do aplicativo e a importância do projeto para todo o pessoal envolvido.]

3 DESCRIÇÕES DOS ENVOLVIDOS E USUÁRIOS

[Para fornecer, de maneira eficiente, produtos e serviços que atendam às reais necessidades dos usuários e dos envolvidos, é necessário identificar e considerar todos os envolvidos como parte do processo de Modelagem de Requisitos. É necessário também identificar os usuários do sistema e assegurar que a comunidade de envolvidos os represente adequadamente. Esta seção fornece um perfil dos envolvidos e dos usuários que integram o projeto, e dos principais problemas que, de acordo com o ponto de vista deles, poderão ser abordados pela solução proposta. Ela não descreve as solicitações ou os requisitos específicos dos usuários e dos envolvidos, já que eles são capturados em um artefato individual de solicitações dos envolvidos. Em vez disso, ela fornece a base e a justificativa que explicam por que os requisitos são necessários.]

3.1 DEMOGRAFIA DO MERCADO

[Resuma as principais demografias do mercado que motivam as decisões do produto. Descreva e posicione os segmentos do mercado-alvo. Faça uma estimativa do tamanho e do crescimento dos mercados usando o número de possíveis usuários ou o valor gasto por seus clientes na tentativa de satisfazer necessidades que serão atendidas por seu produto ou melhoria. Revise as principais tendências e tecnologias do setor. Responda a estas perguntas estratégicas:
Qual é a reputação da sua empresa nesses mercados?
Qual você gostaria que fosse?
Como esse produto ou serviço suporta suas metas?]

3.2 RESUMO DOS ENVOLVIDOS

[Há uma série de envolvidos que se interessam pelo desenvolvimento e nem todos eles são usuários finais. Apresente uma lista resumida desses envolvidos que não são usuários. (O resumo dos usuários encontra-se na seção 3.3.)]

Nome	***Descrição***	***Responsabilidades***
[Informe o tipo de envolvidos.]	*Faça uma breve descrição dos envolvidos.]*	*[Resuma as principais responsabilidades do envolvido no que diz respeito ao sistema em desenvolvimento; ou seja, o interesse dele como envolvido. Por exemplo, este envolvido:* *- garante que o sistema terá manutenção* *- garante que haverá uma demanda do mercado para as características do produto* *- monitora o andamento do projeto*

- aprova fundos, etc.]

3.3 RESUMO DOS USUÁRIOS

[Apresente uma lista resumida de todos os usuários identificados.]

Nome	***Descrição***	***Responsabilidades***	***Envolvidos***
[Informe o tipo de envolvidos.]	*Faça uma breve descrição dos envolvidos.]*	*[Resuma as principais responsabilidades do envolvido no que diz respeito ao sistema em desenvolvimento; ou seja, o interesse dele como envolvido. Por exemplo, este envolvido:* *- garante que o sistema terá manutenção* *- garante que haverá uma demanda do mercado para as características do produto* *- monitora o andamento do projeto* *- aprova fundos, etc.]*	*[Se o usuário não estiver representado diretamente, identifique o envolvido responsável por representar o interesse do usuário.]*

3.4 AMBIENTE DO USUÁRIO

[Detalhe o ambiente de trabalho do usuário-alvo. A seguir, são apresentadas algumas sugestões:
Número de pessoas envolvidas na execução da tarefa? Isso está mudando?
Qual é a duração de um ciclo de tarefas? Qual é o tempo gasto em cada atividade? Isso está mudando?
Existem restrições ambientais exclusivas: telefone celular, ambientes ao ar livre, uso em aeronaves e outros?
Quais plataformas de sistema estão sendo utilizadas atualmente? Quais são as futuras plataformas?
Que outros aplicativos estão em uso? É necessário que o seu aplicativo interaja com eles?
Este é local em que podem ser incluídos os extratos do Modelo de Negócios para descrever a tarefa e os papéis envolvidos, etc.]

3.5 PERFIS DOS ENVOLVIDOS

[Descreva aqui cada envolvido no sistema preenchendo a tabela abaixo para cada um deles. Lembre-se de que os tipos de envolvidos poderão ser os mais diversos como, por exemplo, usuários, departamentos e desenvolvedores técnicos. Um perfil completo deve abranger os tópicos abaixo para cada tipo de envolvido.]

3.6 <NOME DO ENVOLVIDO>

Representante [Quem é o representante do envolvido no projeto? (Opcional se já estiver documentado em outro lugar.) Forneça o nome da pessoa.]

Representante	*[Uma breve descrição do tipo envolvido.]*
Tipo	*[Qualifique a habilidade, a formação técnica e o grau de sofisticação do envolvido — ou seja, se ele é um guru, executivo, especialista, usuário eventual e assim por diante.]*
Responsabilidades	*[Liste as principais responsabilidades dos envolvidos no que diz respeito ao sistema em desenvolvimento; ou seja, o interesse deles como envolvidos.]*
Critérios de Sucesso	*[Como o envolvido define sucesso?* *De que forma o envolvido é recompensado?]*
Envolvimento	*[Qual é o grau de comprometimento do envolvido no projeto? Especifique, quando possível, os papéis exercidos no Rational Unified Process — ou seja, Revisor de Requisitos etc.]*
Produtos Liberados	*[Há produtos liberados adicionais necessários ao envolvido? Podem ser os produtos liberados do projeto ou as saídas do sistema em desenvolvimento.]*
Comentários/Problemas	*[Problemas que interfiram no bom andamento do projeto e outras informações relevantes devem ser relacionados aqui.]*

3.7 PERFIS DE USUÁRIOS

[Descreva cada usuário único do sistema preenchendo a seguinte tabela para cada tipo de usuário. Lembre-se de que os tipos de usuário poderão ser os mais diversos como, por exemplo, gurus e principiantes. Por exemplo, um guru poderá precisar de uma ferramenta flexível sofisticada com suporte a plataformas cruzadas, enquanto um principiante poderá precisar de uma ferramenta amigável e de fácil utilização. Um perfil completo abrangerá os seguintes tópicos para cada tipo de usuário.]

3.7.1 <NOME DO USUÁRIO>

Representante	*[Uma breve descrição do tipo envolvido.]*
Tipo	*[Qualifique a habilidade, a formação técnica e o grau de sofisticação do envolvido — ou seja, se ele é um guru, executivo, especialista, usuário eventual e assim por diante.]*
Responsabilidades	*[Liste as principais responsabilidades dos envolvidos no que diz respeito ao sistema em desenvolvimento; ou seja, o interesse deles como envolvidos.]*
Critérios de Sucesso	*[Como o envolvido define sucesso?* *De que forma o envolvido é recompensado?]*
Envolvimento	*[Qual é o grau de comprometimento do envolvido no projeto? Especifique, quando possível, os papéis exercidos no Rational Unified Process — ou seja, Revisor de Requisitos etc.]*
Produtos Liberados	*[Há produtos liberados adicionais necessários ao envolvido? Podem ser os produtos liberados do projeto ou as saídas do sistema em desenvolvimento.]*
Comentários/Problemas	*[Problemas que interfiram no bom andamento do projeto e outras informações relevantes devem ser relacionados aqui.]*

3.8 RINCIPAIS NECESSIDADES DOS USUÁRIOS OU DOS ENVOLVIDOS

[Liste os principais problemas com as soluções existentes conforme o ponto de vista do envolvido ou do usuário. Para cada problema, esclareça os seguintes pontos:
Quais são as causas do problema?
Como ele está sendo resolvido agora?
Que soluções o envolvido ou usuário deseja?]
[É essencial entender a importância relativa atribuída pelo envolvido ou usuário à resolução de cada problema. A s técnicas de ordenação e de votação cumulativa indicam os problemas que devem ser resolvidos versus os problemas que eles gostariam que fossem resolvidos.
Preencha a tabela a seguir — se estiver usando o Rational RequisitePro para capturar as Necessidades, pode ser um fragmento ou relatório dessa ferramenta.]

Necessidade	Prioridade	Preocupações	Solução atual	Soluções propostas

3.9 ALTERNATIVAS E CONCORRÊNCIA

[Identifique as alternativas que o envolvido considera disponíveis. Isso inclui adquirir um produto do concorrente, desenvolver uma solução própria ou simplesmente manter o estado atual. Liste as opções conhecidas que a concorrência oferece ou que podem se tornar disponíveis. Inclua os principais pontos fortes e pontos fracos de cada concorrente segundo o ponto de vista do envolvido ou do usuário final.]

3.9.1 <UMCOMPETIDOR>

3.9.2 <OUTROCOMPETIDOR>

4 VISÃO GERAL DO PRODUTO

[Esta seção oferece uma visão de nível superior dos recursos do produto, interfaces com outros aplicativos e configurações de sistema. Ela geralmente é constituída destas três subseções:
Perspectiva do produto
Funções do produto
Suposições e dependências]

4.1 PERSPECTIVA DO PRODUTO

[Esta subseção do documento de Visão coloca o produto na perspectiva de outros produtos relacionados e do ambiente do usuário. Se o produto for independente e totalmente auto-suficiente, exponha isso aqui. Se o produto for um componente de um sistema maior, esta subseção deverá relacionar como esses sistemas interagem e identificar as interfaces relevantes entre os sistemas. Uma maneira fácil de exibir os principais componentes do sistema maior, suas interconexões e interfaces externas é através de um diagrama de bloco.]

4.2 RESUMO DOS RECURSOS

[Resuma os principais benefícios e recursos que o produto fornecerá. Por exemplo, um documento Visão referente a um sistema de suporte ao cliente poderá usar esta seção para abordar a documentação de problemas, o roteamento e a elaboração de relatórios de status sem mencionar a quantidade de detalhes necessária a cada uma dessas funções.
Organize as funções de modo que a lista possa ser compreendida pelo cliente ou por qualquer pessoa que esteja lendo o documento pela primeira vez. Uma tabela simples relacionando os principais benefícios e seus recursos de suporte poderá ser suficiente. Por exemplo:]

Recursos de suporte	*Benefícios para o cliente*
Novo recurso 1	*Benefício que ele trará pra o cliente.*
Novo recurso 2	*Benefício que ele trará para o cliente.*

4.3 SUPOSIÇÕES E DEPENDÊNCIAS

[Liste cada fator que afeta os recursos especificados no documento Visão. Liste as suposições que, se forem mudadas, alterarão o documento de Visão. Por exemplo, uma suposição poderá estabelecer que um sistema operacional específico estará disponível para o hardware projetado para o produto de software. Se o sistema operacional não estiver disponível, o documento de Visão deverá ser mudado.]

5 RECURSOS DO PRODUTO

[Liste e descreva brevemente os recursos do produto. Trata-se dos recursos de nível superior do sistema que são necessários para propiciar benefícios aos usuários. Cada recurso é um serviço desejado externamente que normalmente exige uma série de entradas para alcançar os resultados desejados. Por exemplo, um dos recursos de um sistema de rastreamento de problemas poderá ser a capacidade de fornecer relatórios de tendências. À medida que o modelo de casos de uso for desenvolvido, atualize a descrição para fazer referência aos casos de uso.

Como o documento de Visão é revisado por muitas pessoas envolvidas, o nível de detalhes deve ser geral o suficiente para que todos entendam. No entanto, devem estar disponíveis detalhes suficientes para fornecer à equipe as informações necessárias para criar um modelo de casos de uso.

Para gerenciar a complexidade dos aplicativos de maneira eficiente, é recomendável para qualquer sistema novo, ou para uma adição que complemente um sistema existente, que seja utilizado um grau de abstração de nível suficientemente elevado de modo a resultar em 25 a 99 recursos. Esses recursos serão a base fundamental do gerenciamento do projeto, do gerenciamento do escopo e da definição do produto. Cada recurso será descrito mais detalhadamente no modelo de casos de uso.

Em toda esta seção, cada recurso será percebido externamente por usuários, operadores ou outros sistemas externos. Esses recursos deverão incluir uma descrição da funcionalidade e de todas as questões de usabilidade relevantes que deverão ser abordadas. As seguintes diretrizes se aplicam:

5.1 <UM RECURSO>

5.2 <OUTRO RECURSO>

6 RESTRIÇÕES

[Mencione quaisquer restrições de design, restrições externas ou outras dependências.]

7 FAIXAS DE QUALIDADE

[Defina as faixas de qualidade para desempenho, robustez, tolerância a erros, usabilidade e características semelhantes que não são capturadas no Conjunto de Recursos.]

8 PRECEDÊNCIA E PRIORIDADE

[Defina a prioridade dos diferentes recursos do sistema.]

9 OUTROS REQUISITOS DO PRODUTO

[Em um nível alto, liste os padrões aplicáveis, os requisitos de hardware ou de plataforma, os requisitos de desempenho e os requisitos de ambiente.]

9.1 PADRÕES APLICÁVEIS

[Liste todos os padrões com os quais o produto deverá estar em conformidade. Entre eles, poderão estar incluídos padrões legais e reguladores (FDA, UCC), padrões de comunicações (TCP/IP, ISDN), padrões de conformidade com plataformas (Windows, UNIX etc) e padrões de qualidade e de segurança (UL, ISO, CMM).]

9.2 REQUISITOS DO SISTEMA

[Defina todos os requisitos do sistema necessários para suportar o aplicativo. Entre eles, poderão estar incluídos as plataformas de rede e os sistemas de operacionais de host suportados, configurações, memória, periféricos e software fornecido.]

9.3 REQUISITOS DE DESEMPENHO

[Use esta seção para detalhar os requisitos de desempenho. Os problemas de desempenho podem abranger itens como fatores de carga do usuário, largura de banda ou capacidade de comunicação, taxa de transferência, precisão e confiabilidade ou tempos de resposta em uma série de condições de carregamento.]

9.4 REQUISITOS AMBIENTAIS

[Detalhe os requisitos ambientais, conforme necessário. Para sistemas baseados em hardware, as questões ambientais poderão incluir temperatura, choques, umidade, radiação etc. Para aplicativos de software, os fatores ambientais podem incluir condições de uso, ambiente do usuário, disponibilidade de recursos, problemas de manutenção, e recuperação e tratamento de erros.]

10 REQUISITOS DA DOCUMENTAÇÃO

[Esta seção descreve a documentação que deverá ser desenvolvida para suportar a implantação bem-sucedida de aplicativos.]

10.1 MANUAL DO USUÁRIO

[Descreva a finalidade e o conteúdo do Manual do Usuário. Discuta questões como o tamanho desejado, o nível de detalhamento, a necessidade de um índice, o uso de um glossário de termos, estratégia de tutorial versus de manual de referência etc. As restrições de formatação e de impressão também deverão ser identificadas.]

10.2 AJUDA ON-LINE

[Muitos aplicativos fornecem um sistema de ajuda on-line para auxiliar o usuário. A natureza desses sistemas é exclusiva do desenvolvimento do aplicativo já que eles combinam aspectos de programação (hyperlinks etc) com aspectos de escrita técnica como, por exemplo, organização e apresentação. Muitos perceberam que o desenvolvimento de um sistema de ajuda on-line é um projeto que está contido em outro projeto, beneficiando-se do gerenciamento adiantado do escopo e da atividade de planejamento.]

10.3 GUIAS DE INSTALAÇÃO E DE CONFIGURAÇÃO, E ARQUIVO LEIAME

[Um documento que inclua instruções de instalação e diretrizes de configuração é importante para se oferecer uma solução completa. Além disso, um arquivo Leiame é normalmente incluído como um componente padrão. O arquivo Leiame poderá incluir uma seção "O Que Há de Novo Neste Release" e uma discussão dos problemas de compatibilidade em relação aos releases anteriores. A maior parte dos usuários também considera desejável que o arquivo Leiame documente erros e soluções conhecidos.]

10.4 ROTULAÇÃO E EMBALAGEM

[Os aplicativos modernos atuais apresentam uma aparência consistente que é percebida inicialmente na embalagem do produto e se propaga pelos menus de instalação, telas iniciais, sistemas de ajuda, caixas de diálogo GUI etc. Esta seção define as necessidades e os tipos de rotulação a serem incorporados no código. Como exemplos, podemos citar observações sobre direitos autorais e patentes, logotipos corporativos, ícones padronizados, outros elementos gráficos etc.]

11 ATRIBUTOS DE RECURSOS

[São designados atributos para os recursos que podem ser usados para avaliar, rastrear, priorizar e gerenciar os itens do produto cuja implementação foi proposta. Todos os atributos e tipos de requisitos devem ser descritos no Plano de Gerenciamento de Requisitos. No entanto, talvez seja conveniente listar e descrever brevemente os atributos referentes aos recursos que foram escolhidos. As subseções a seguir representam um conjunto de atributos de recursos sugeridos.]

11.1 STATUS

[Definido pela equipe de gerenciamento do projeto após a negociação e a revisão. Controla o andamento durante a definição da baseline do projeto.]

Proposto	***[Usado para descrever recursos que estão sendo discutidos, mas que ainda não foram revisados e aceitos pelo "canal oficial" como, por exemplo, um grupo de trabalho formado por representantes da equipe do projeto, do gerenciamento do produto e da comunidade de usuários ou de clientes.]***
Aprovado	*[Recursos que são considerados úteis e viáveis, e que foram aprovados para implementação pelo canal oficial.]*
Incorporado	*[Recursos incorporados à baseline do produto em um momento específico no tempo.]*

11.2 BENEFÍCIO

[Definido pelo departamento de marketing, pelo gerente do produto ou pelo analista de negócios. Todos os requisitos diferem entre si. Classificar os requisitos por seu benefício relativo para o usuário final dá início a um diálogo com os clientes, analistas e membros da equipe de desenvolvimento. Usado no gerenciamento do escopo e na determinação da prioridade de desenvolvimento.]

11.3 ESFORÇO

[Definido pela equipe de desenvolvimento. Como algumas funcionalidades necessitam de mais tempo e de mais recursos do que outras, estimar o número de semanas de participação de cada pessoa ou equipe, as linhas de código necessárias ou os pontos de função, por exemplo, é a melhor maneira de avaliar a complexidade e definir expectativas do que poderá ou não ser feito em um determinado período de tempo. Usado no gerenciamento do escopo e na determinação da prioridade de desenvolvimento.]

11.4 RISCO

[Definido pela equipe de desenvolvimento com base na probabilidade de ocorrerem eventos indesejáveis no projeto como, por exemplo, custos excessivos, atrasos na programação ou até cancelamentos. A maior parte dos gerentes de projeto considera que a categorização dos riscos em altos, médios e baixos é suficiente, embora sejam possíveis gradações ainda mais específicas. Freqüentemente os riscos poderão ser avaliados indiretamente medindo-se o grau de incerteza (intervalo) da estimativa de programação da equipe dos projetos.]

11.5 RELEASE-ALVO

[Registra a versão planejada do produto em que o recurso aparecerá pela primeira vez. Este campo pode ser usado para alocar recursos de um documento de visão em um release de baseline específico. Quando for usado em conjunto com o campo de status, sua equipe poderá propor, registrar e discutir vários recursos do release sem que tenham que ser necessariamente desenvolvidos. Somente serão implementados os recursos cujo Status estiver definido como Incorporado e cujo Release-alvo estiver definido. Quando ocorrer o gerenciamento do escopo, o Número da Versão do Release-alvo poderá ser aumentado de modo que o item permaneça no documento Visão, mas seja programado para um release posterior.]

APÊNDICE C.3 – GLOSSÁRIO

<Nome da Empresa>
<Slogan>

<Rua>, Nº <Número> - <Bairro> - <Cidade> - <UF>
CEP: <CEP>

<NOME DO PROJETO>

GLOSSÁRIO

VERSÃO 1

Histórico de revisão

Data	Versão	Descrição	Autor

SUMÁRIO

1 GLOSSÁRIO 143
1.1 INTRODUÇÃO 143
1.2 FINALIDADE 143
1.3 ESCOPO 143
1.4 REFERÊNCIAS 143
1.5 VISÃO GERAL 143
2 DEFINIÇÕES 143
2.1 <UM TERMO> 144
2.2 <OUTRO TERMO> 144
2.3 <UM GRUPO DE TERMOS> 144
2.3.1 <Um sub-grupo de termo> 144
2.3.2 <Outro sub-grupo de termo> 144
2.4 < UM SEGUNDO GRUPO DE TERMOS> 145
2.4.1 < Mais um sub-grupo de termos> 145
2.4.2 <Outro sub-grupo de termo> 145

1 GLOSSÁRIO

1.1 INTRODUÇÃO

[A introdução do Glossário fornece uma visão geral de todo o documento. Apresente todas as informações que poderão ser necessárias para que o leitor compreenda o documento nesta seção. Este documento é usado para definir a terminologia específica do domínio do problema, explicando termos que podem ser desconhecidos do leitor das descrições de caso de uso ou de outros documentos do projeto. Freqüentemente, este documento poderá ser usado como um dicionário de dados informal, capturando definições de dados para que as descrições de caso de uso e outros documentos do projeto possam concentrar-se no que o sistema deve fazer com as informações. Este documento deverá ser salvo em um arquivo denominado Glossário.]

1.2 FINALIDADE

[Especifique a finalidade deste Glossário.]

1.3 ESCOPO

[Uma breve descrição do escopo deste Glossário; a que Projeto(s) ele está associado e tudo o mais que seja afetado ou influenciado por este documento.]

1.4 REFERÊNCIAS

[Esta subseção fornece uma lista completa de todos os documentos mencionados em qualquer outra parte do Glossário. Identifique cada documento por título, número do relatório (se aplicável), data e organização de publicação. Especifique as fontes a partir das quais as referências podem ser obtidas. Essas informações podem ser fornecidas por um anexo ou outro documento.]

1.5 VISÃO GERAL

[Esta subseção descreve o que o restante do Glossário contém e explica como o documento está organizado.]
Definições

[Os termos definidos aqui são a essência do documento. Eles poderão ser definidos na ordem desejada, mas geralmente a ordem alfabética propicia maior acessibilidade.]

1.6 <UM TERMO>

[A definição de <Um termo>é apresentada aqui. Você deverá apresentar quantas informações forem necessárias para que o leitor compreenda o conceito.]

1.7 <OUTRO TERMO>

[A definição de <Outro termo> é apresentada aqui. Você deverá apresentar quantas informações forem necessárias para que o leitor compreenda o conceito]

1.8 <UM GRUPO DE TERMOS>

[Às vezes, é útil organizar os termos em grupos para aprimorar a legibilidade. Por exemplo, se o domínio do problema contiver termos relacionados a contabilidade e a construção de prédios (como seria o caso se estivéssemos desenvolvendo um sistema para gerenciar projetos de construção), apresentar os termos dos dois subdomínios diferentes poderá gerar confusão para o leitor. Para resolver o problema, utilizamos agrupamentos de termos. Ao apresentar os agrupamentos de termos, forneça uma breve descrição que ajude o leitor a compreender o que <Um grupo de termos> representa. Os termos apresentados no grupo deverão ser organizados alfabeticamente para possibilitar um fácil acesso.]

1.8.1 <Um sub-grupo de termo>

[A definição de <Um sub-grupo de termo> é apresentada aqui. Apresente quantas informações forem necessárias para que o leitor compreenda o conceito.]

1.8.2 <Outro sub-grupo de termo>

[A definição de <Outro sub-grupo de termo> é apresentada aqui. Apresente quantas informações forem necessárias para que o leitor compreenda o conceito.]

1.9 < UM SEGUNDO GRUPO DE TERMOS>

1.9.1 < Mais um sub-grupo de termos>

[A definição do termo é apresentada aqui. Apresente quantas informações forem necessárias para que o leitor compreenda o conceito.]

1.9.2 <Outro sub-grupo de termo>

[A definição do termo é apresentada aqui. Apresente quantas informações forem necessárias para que o leitor compreenda o conceito.]

APÊNDICE C.4 – REGRAS DE NEGÓCIO

<Nome da Empresa>
<Slogan>

<Rua>, Nº <Número> - <Bairro> - <Cidade> - <UF>
CEP: <CEP>

<NOME DO PROJETO>

REGRAS DE NEGÓCIO

VERSÃO 1

Histórico de revisão

Data	Versão	Descrição	Autor

SUMÁRIO

1 REGRAS DE NEGÓCIOS 155
1.1 INTRODUÇÃO 155
1.2 FINALIDADE 155
1.3 ESCOPO 155
1.4 REFERÊNCIAS 155
1.5 VISÃO GERAL 155
1.6 DEFINIÇÕES 155
2 REGRAS 156
2.1 [RN001] <REGRAS DE NEGÓCIO> 156
2.2 [RN002] <OUTRAS REGRAS DE NEGÓCIO> 156
2.3 [GRN001] <UM GRUPO DE REGRAS DE NEGÓCIO> 156
2.3.1 [RN0003] <Uma regra de negócio do grupo> 156
2.3.2 [RN0004] <Outra regra de negócio do grupo> 156
2.4 [GRN002] <UM SEGUNDO GRUPO DE REGRAS DE NEGÓCIO> 157
2.4.1 [RN0005] <Mais uma regra de negocio do grupo> 157
2.4.2 [RN0006] <Uma outra regra de negócio de grupo> 157

2 REGRAS DE NEGÓCIOS

2.1 INTRODUÇÃO

[A introdução das Regras de Negócios oferece uma visão geral de todo o documento. Apresente todas as informações de que o leitor pode precisar para entender o documento nesta seção. Salve este documento em um arquivo denominado Regras de Negócios.]

2.2 FINALIDADE

[Especifique a finalidade deste documento.]

2.3 ESCOPO

[Uma breve descrição do escopo do documento Regras de Negócios; o(s) Projeto(s) ao(s) qual(is) ele está associado e tudo o que é afetado ou influenciado por este documento.]

2.4 REFERÊNCIAS

[Esta subseção apresenta uma lista completa de todos os documentos mencionados no documento Regras de Negócios. Identifique cada documento por título, número do relatório (se aplicável), data e organização de publicação. Especifique as fontes a partir das quais as referências podem ser obtidas. Essas informações podem ser fornecidas por um anexo ou outro documento.]

2.5 VISÃO GERAL

[Esta subseção descreve o conteúdo restante do documento Regras de Negócios e explica como ele está organizado.]

2.6 DEFINIÇÕES

[Os termos definidos aqui formam a parte essencial do documento. Eles podem ser definidos na ordem desejada, mas geralmente a ordem alfabética proporciona maior acessibilidade.]

3 REGRAS

3.1 [RN001] <REGRAS DE NEGÓCIO>

[A definição de <regras de negócio> é apresentada aqui, com todas as informações necessárias para que o leitor entenda o conceito.]

3.2 [RN002] <OUTRAS REGRAS DE NEGÓCIO>

[A definição de <Outras regras de negócio> é apresentada aqui, com todas as informações necessárias para que o leitor entenda o conceito.]

3.3 [GRN001] <UM GRUPO DE REGRAS DE NEGÓCIO>

[Às vezes é útil organizar as Regras de Negócios em grupos para melhorar a leitura. Por exemplo, se o domínio de problema contém Regras de Negócios relacionadas a contabilidade e construção civil (como seria o caso se estivéssemos desenvolvendo um sistema para gerenciar projetos de construção), a apresentação das Regras de Negócios dos dois subdomínios diferentes pode ser confusa para o leitor. Para resolver esse problema, utilizamos grupos de Regras de Negócios. Ao apresentar os grupos de Regras de Negócios, forneça uma pequena descrição que ajude o leitor a entender o que <Um grupo de regras de negócio> representa. As Regras de Negócios apresentadas no grupo são organizadas em ordem alfabética para facilitar o acesso.]

3.3.1 [RN0003] <Uma regra de negócio do grupo>

[A definição de <uma regra de negócio do grupo> é apresentada aqui, com todas as informações necessárias para que o leitor entenda o conceito.]

3.3.2 [RN0004] <Outra regra de negócio do grupo>

[A definição de <Outra regra de negócio do grupo> é apresentada aqui, com todas as informações necessárias para que o leitor entenda o conceito.]

3.4 [GRN002] <UM SEGUNDO GRUPO DE REGRAS DE NEGÓCIO>

3.4.1 [RN0005] <Mais uma regra de negocio do grupo>

[A definição de <Mais uma regra de negócio > é apresentada aqui, com todas as informações necessárias para que o leitor entenda o conceito.]

3.4.2 [RN0006] <Uma outra regra de negócio de grupo>

[A definição de <Uma outra regra de negócio do grupo> é apresentada aqui, com todas as informações necessárias para que o leitor entenda o conceito.]

APÊNDICE C.5 – ATA DE REUNIÃO

[ASSUNTO]

[TÍTULO]

[DATA]	**[HORÁRIO]**	**LOCAL**

Reunião presidida por:	*<Nome do presidente da reunião>*
Tipo de reunião:	*<Estratégica | Decisória | Informativa | Negocial>*
Facilitador(a):	*<Nome do facilitador(a)>*
Secretário(a):	*<Nome do secretário(a)>*
Cronometrista:	*<Nome do(a) cronometrista>*
Participantes:	*<Participante 1>* *<Participante 2>* *<Participante n>*

TÓPICOS DA AGENDA

- **[PRIMEIRO TÓPICO DA AGENDA]**
- **[SEGUNDO TÓPICO DA AGENDA]**
- **[ENÉSIMO TÓPICO DA AGENDA]**

[TEMPO ALOCADO]	**[PRIMEIRO TÓPICO DA AGENDA]**	**[APRESENTADOR]**
Discussão	*<Nome do presidente da reunião>*	
Conclusões	*<Estratégica | Decisória | Informativa | Negocial>*	

Itens de ação	**Responsável**	**Prazo**
<Primeira ação a ser executada>	*<Nome do responsável>*	*<dd/mm/aaaa>*
<Segunda ação a ser executada>	*<Nome do responsável>*	*<dd/mm/aaaa>*
<Enésima ação a ser executada>	*<Nome do responsável>*	*<dd/mm/aaaa>*

[TEMPO ALOCADO]	**[SEGUNDO TÓPICO DA AGENDA]**	**[APRESENTADOR]**
Discussão	*<Nome do presidente da reunião>*	
Conclusões	*<Estratégica | Decisória | Informativa | Negocial>*	

Itens de ação	Responsável	Prazo
<Primeira ação a ser executada>	*<Nome do responsável>*	*<dd/mm/aaaa>*
<Segunda ação a ser executada>	*<Nome do responsável>*	*<dd/mm/aaaa>*
<Enésima ação a ser executada>	*<Nome do responsável>*	*<dd/mm/aaaa>*

Observadores	*<Nome do presidente da reunião>*
Recursos	*<Estratégica | Decisória | Informativa | Negocial>*
Observações especiais	
Leitura necessária	

12 de março de 2010, _____

Assinatura do presidente da reunião

APÊNDICE C.6 – DOCUMENTO DE ESPECIFICAÇÃO DE REQUISITOS

<Nome da Empresa>
<Slogan>

<Rua>, Nº <Número> - <Bairro> - <Cidade> - <UF>
CEP: <CEP>

<NOME DO PROJETO>

DOCUMENTO DE ELICITAÇÃO DE REQUISITOS

VERSÃO 1

Histórico de revisão

Data	Versão	Descrição	Autor

SUMÁRIO

1 ESPECIFICAÇÃO DOS REQUISITOS DO SOFTWARE 171
1.1 INTRODUÇÃO 171
1.2 FINALIDADE 171
1.3 ESCOPO 171
1.4 DEFINIÇÕES, ACRÔNIMOS E ABREVIAÇÕES 171
1.5 REFERÊNCIAS 172
1.6 VISÃO GERAL 172
2 REQUISITOS FUNCIONAIS 173
2.1 [RF001] NOME DO PRIMEIRO REQUISITO 173
2.2 [RF002] NOME OUTRO REQUISITO 173
3 REQUISITOS NÃO-FUNCIONAIS 174
3.1 USABILIDADE 174
3.1.1 [RNF001] Primeiro requisito não-funcional de usabilidade 174
3.1.2 [RNF002] Segundo requisito não-funcional de usabilidade 174
3.2 CONFIABILIDADE 174
3.2.1 [RNF003] Primeiro requisito não-funcional de confiabilidade 175
3.2.2 [RNF004] Segundo requisito não-funcional de confiabilidade 175
3.3 DESEMPENHO 175
3.3.1 [RNF005] Primeiro requisito não-funcional de desempenho 175
3.3.2 [RNF006] Segundo requisito não-funcional de desempenho 175
3.4 INTERFACES 175
3.4.1 Interfaces do Usuário 176
3.4.2 Interfaces de Hardware 176
3.4.3 Interfaces de Software 176
3.4.4 Interfaces de Comunicação 176
4 ESCOPO NEGATIVO 177

1 ESPECIFICAÇÃO DOS REQUISITOS DO SOFTWARE

1.1 INTRODUÇÃO

[A introdução da Especificação de Requisitos de Software (SRS) fornece uma visão geral de toda a SRS. Ela contém a finalidade, o escopo, as definições, os acrônimos, as abreviações, as referências e a visão geral da SRS.]
[Observação: A SRS captura todos os requisitos de software do sistema ou de uma parte do sistema. A seguir, há um esquema de uma SRS típica para um projeto que utiliza somente requisitos em estilo de linguagem natural tradicional — sem modelagem de casos de uso. Essa SRS captura todos os requisitos em um único documento, com seções aplicáveis inseridas a partir das Especificações Suplementares (que não serão mais necessárias). Para ter acesso a um template de uma SRS que utilize a modelagem de casos de uso, que consiste em um pacote contendo Casos de Uso do modelo de casos de uso e Especificações Suplementares aplicáveis, assim como outras informações de suporte, consulte o arquivo rup_srsuc.dot.]
[É possível organizar a SRS de várias maneiras diferentes. Consulte o padrão [IEEE830-1998] para obter explicações mais detalhadas, assim como outras opções de organização de uma SRS.]

1.2 FINALIDADE

[Especifique a finalidade desta SRS. A SRS descreve totalmente o comportamento externo do aplicativo ou do subsistema identificado. Ela também descreve requisitos não funcionais, restrições de design e outros fatores necessários para fornecer uma visão completa e abrangente dos requisitos do software.]

1.3 ESCOPO

[Uma breve descrição do aplicativo de software ao qual se aplica a SRS, do recurso ou de outro agrupamento de subsistemas, do(s) modelo(s) de Casos de Uso associado(s) a ela e de tudo o que for afetado ou influenciado por este documento.]

1.4 DEFINIÇÕES, ACRÔNIMOS E ABREVIAÇÕES

[Esta subseção fornece as definições de todos os termos, acrônimos e abreviações necessárias à adequada interpretação da SRS. Essas informações podem ser fornecidas mediante referência ao Glossário do projeto.]

1.5 REFERÊNCIAS

[Esta subseção fornece uma lista completa de todos os documentos mencionados em qualquer outra parte da SRS. Identifique cada documento por título, número do relatório (se aplicável), data e organização de publicação. Especifique as fontes a partir das quais as referências podem ser obtidas. Essas informações podem ser fornecidas por um anexo ou outro documento.]

1.6 VISÃO GERAL

[Esta subseção descreve o que o restante da SRS contém e explica como o documento está organizado.]

2 REQUISITOS FUNCIONAIS

[Requisitos funcionais de um software indicam as funcionalidades desenvolvidas visando a resolução de um problema. Neste documento eles são identificados pelo prefixo "RF" seguido de um número seqüencial, entre colchetes, como por exemplo, [RF0001]. Os identificadores dos requisitos não devem ser reiniciados a cada subseção. Este número sequencial não deve ser modificado ou reaproveitado, para não invalidar referências externas feitas a eles.]

2.1 [RF001] NOME DO PRIMEIRO REQUISITO

[Uma descrição detalhada do requisito RF001]

Tamanho	Pior caso:	Caso normal:	Melhor caso:
Prioridade	☐ Essencial	☐ Importante	☐ Desejável

2.2 [RF002] NOME OUTRO REQUISITO

[Uma descrição detalhada do requisito RF002]

Tamanho	Pior caso:	Caso normal:	Melhor caso:
Prioridade	☐ Essencial	☐ Importante	☐ Desejável

3 REQUISITOS NÃO-FUNCIONAIS

[Aqui serão descritos os requisitos não funcionais, suas características e sua importância para o projeto]

3.1 USABILIDADE

[Esta seção contém todos os requisitos que afetam a usabilidade. Por exemplo:

- *Especifique o tempo de treinamento necessário para que usuários normais e usuários com conhecimentos avançados se tornem produtivos em operações específicas*
- *Especifique períodos de tempo mensuráveis para tarefas típicas ou baseie os requisitos de usabilidade do novo sistema em outros sistemas que os usuários conheçam e gostem*
- *Especifique requisitos de forma que estejam em conformidade com padrões de usabilidade comuns como, por exemplo, os padrões CUA da IBM ou os padrões GUI da Microsoft]*

3.1.1 [RNF001] Primeiro requisito não-funcional de usabilidade

[Uma descrição detalhada do requisito RNF001]

3.1.2 [RNF002] Segundo requisito não-funcional de usabilidade

[Uma descrição detalhada do requisito RNF001]

3.2 CONFIABILIDADE

[Os requisitos de confiabilidade do sistema devem ser especificados aqui. A seguir, há algumas sugestões:

- *Disponibilidade — especifique a porcentagem de tempo disponível (xx.xx%), as horas de uso, o acesso à manutenção, as operações de modo degradado, etc.*
- *Tempo Médio entre Falhas (MTBF) — normalmente especificado em horas, mas também poderá ser especificado em termos de dias, meses ou anos.*
- *Tempo Médio para Reparo (MTTR) — quanto tempo o sistema poderá ficar sem funcionar após uma falha?*
- *Exatidão — especifique a precisão (resolução) e a exatidão (através de algum padrão conhecido) necessárias na saída do sistema.*
- *Taxa Máxima de Erros ou Defeitos — geralmente expressa em termos de erros por milhares de linhas de código (erros/KLOC) ou de erros por ponto de função (erros/ponto de função).*

- *Taxa de Erros ou Defeitos — categorizada em termos de erros pouco importantes, importantes e críticos: o(s) requisito(s) deve(m) definir o que se entende por um erro "crítico"; por exemplo, a perda total de dados ou uma total incapacidade de usar determinadas partes da funcionalidade do sistema.]*

3.2.1 [RNF003] Primeiro requisito não-funcional de confiabilidade

[Uma descrição detalhada do requisito RNF003]

3.2.2 [RNF004] Segundo requisito não-funcional de confiabilidade

[Uma descrição detalhada do requisito RNF004]

3.3 DESEMPENHO

[As características de desempenho do sistema devem ser descritas nesta seção. Inclua tempos de resposta específicos. Quando aplicável, faça referência, por nome, aos Casos de Uso relacionados.

- *Tempo de resposta de uma transação (médio, máximo)*
- *Taxa de transferência como, por exemplo, transações por segundo*
- *Capacidade como, por exemplo, o número de clientes ou de transações que o sistema pode acomodar*
- *Modos de degradação (o modo aceitável de operação quando o sistema tiver sido degradado de alguma maneira)*
- *A utilização de recursos como, por exemplo, memória, disco, comunicações, etc.*

3.3.1 [RNF005] Primeiro requisito não-funcional de desempenho

[Uma descrição detalhada do requisito RNF005]

3.3.2 [RNF006] Segundo requisito não-funcional de desempenho

[Uma descrição detalhada do requisito RNF006]

3.4 INTERFACES

[Esta seção define as interfaces que devem ser suportadas pelo aplicativo. Ela deve conter especificidades, protocolos, portas e endereços lógicos adequados, entre outros, para que o software possa ser desenvolvido e verificado em relação aos requisitos de interface.]

3.4.1 Interfaces do Usuário

[Descreva as interfaces de usuário que deverão ser implementadas pelo software.]

3.4.2 Interfaces de Hardware

[Esta seção define todas as interfaces de hardware que devem ser suportadas pelo software, incluindo a estrutura lógica, os endereços físicos, o comportamento esperado, etc.]

3.4.3 Interfaces de Software

[Esta seção descreve as interfaces de software para outros componentes do sistema de software. Poderão ser componentes comprados, componentes reutilizados de outro aplicativo ou componentes que estejam sendo desenvolvidos para subsistemas fora do escopo desta SRS, mas com os quais esse aplicativo de software deve interagir.]

3.4.4 Interfaces de Comunicação

[Descreva todas as interfaces de comunicação com outros sistemas ou dispositivos como, por exemplo, redes locais, dispositivos seriais remotos, etc.]

4 ESCOPO NEGATIVO

[Aqui será descrito todos os requisitos ou características que o produto do projeto não terá. Serão definidas suas características e o reflexo das mesmas no produto final]

APÊNDICE C.7 – DOCUMENTO DE ESPECIFICAÇÃO DE CASO DE USO

<Nome da Empresa>
<Slogan>

<Rua>, Nº <Número> - <Bairro> - <Cidade> - <UF>
CEP: <CEP>

<NOME DO PROJETO>

DOCUMENTO DE ESPECIFICAÇÃO DO CASO DE USO <NOME DO CASO DE USO>

VERSÃO 1

Histórico de revisão

Data	Versão	Descrição	Autor

SUMÁRIO

ESPECIFICAÇÃO DE CASO DE USO: <NOME DO CASO DE USO> 187
1 NOME DO CASO DE USO 188
1.1 BREVE DESCRIÇÃO 188
2 FLUXO DE EVENTOS 189
2.1 FLUXO BÁSICO 189
2.2 FLUXOS ALTERNATIVOS 189
2.2.1 < Primeiro fluxo alternativo > 189
2.2.2 < Um subfluxo alternativo >.....189
2.2.3 < Segundo fluxo alternativo > 190
3 REQUISITOS ESPECIAIS 191
3.1 < PRIMEIRO REQUISITO ESPECIAL > 191
4 PRECONDIÇÕES 192
4.1 < PRECONDIÇÃO UM > 192
5 PÓS-CONDIÇÕES.....193
5.1 < PÓS-CONDIÇÃO UM >.....193
6 PONTOS DE EXTENSÃO.....194
6.1 <NOME DO PONTO DE EXTENSÃO> 194

ESPECIFICAÇÃO DE CASO DE USO: <NOME DO CASO DE USO>

[O template a seguir é fornecido para uma especificação de caso de uso, que contém as propriedades textuais do caso de uso. Os diagramas de caso de uso podem ser desenvolvidos em uma ferramenta de modelagem visual, assim como um relatório de caso de uso, com todas as suas propriedades.]

1 NOME DO CASO DE USO

1.1 BREVE DESCRIÇÃO

[A descrição relata brevemente o papel e a finalidade do caso de uso.]

2 FLUXO DE EVENTOS

2.1 FLUXO BÁSICO

[Este caso de uso é iniciado quando o ator pratica alguma ação. Os casos de uso sempre são iniciados por atores. O caso de uso descreve o que o ator faz e o que o sistema faz em resposta. Ele deve ser elaborado como um diálogo entre o ator e o sistema.
caso de uso descreve o que ocorre no sistema, mas não como ou por quê. Se forem trocadas informações, seja específico no que diz respeito ao conteúdo que é passado e retornado. Por exemplo, não é muito esclarecedor dizer que o ator fornece informações do cliente. É melhor dizer que ele fornece o nome e o endereço do cliente. É útil fazer uso de um Glossário de Termos para manter a complexidade do caso de uso sob controle — poderá ser conveniente definir termos como, por exemplo, informações do cliente neste glossário, a fim de evitar que o caso de uso fique repleto de detalhes.
As alternativas simples poderão ser apresentadas no texto do caso de uso. Se forem necessárias apenas algumas frases para descrever o que acontece quando há uma alternativa, faça essa descrição diretamente na seção fluxo de eventos. Se o fluxo alternativo for mais complexo, use uma seção separada para descrevê-lo. Por exemplo, uma subseção fluxo alternativo explica como descrever alternativas mais complexas.
Às vezes, uma figura vale por mil palavras, embora não haja nada que possa substituir uma redação clara e organizada. Se for mais esclarecedor, sinta-se à vontade para colar representações gráficas de interfaces do usuário, fluxos de processo ou outras imagens no caso de uso. Se um fluxograma for útil para apresentar um processo complexo de decisões, utilize-o sem dúvida nenhuma! Assim como no caso de comportamentos dependentes de estado, um diagrama de transição de estado geralmente esclarece o comportamento de um sistema muito mais do que páginas e páginas de texto. Use o meio de apresentação certo para o problema, mas procure evitar o uso de terminologia, notações ou imagens que o público possa não entender. Lembre-se de que sua finalidade é esclarecer e não obscurecer.]

2.2 FLUXOS ALTERNATIVOS

2.2.1 <Primeiro fluxo alternativo>

[As alternativas mais complexas são descritas em uma seção separada, mencionada na subseção fluxo básico da seção fluxo de eventos. Pense nas subseções fluxo alternativo como comportamentos alternativos — cada fluxo alternativo representa um comportamento alternativo geralmente devido a exceções que ocorrem no fluxo principal. O tamanho desses fluxos poderá ser tão extenso quanto o necessário para descrever os eventos associados ao comportamento alternativo. Quando um fluxo alternativo termina, os eventos do principal fluxo de eventos são retomados, a menos que seja especificado algo em contrário.]

2.2.2 <Um subfluxo alternativo>

[Os fluxos alternativos, por sua vez, podem ser divididos em subseções, se isso contribuir para maior clareza.]

2.2.3 <Segundo fluxo alternativo>

[Pode haver, e muito provavelmente haverá, uma série de fluxos alternativos em um caso de uso. Mantenha cada fluxo alternativo separado para aumentar a clareza. O uso de fluxos alternativos melhora a legibilidade do caso de uso e evita que os casos de uso sejam decompostos em hierarquias de casos de uso. Lembre-se de que os casos de uso são apenas descrições textuais e que sua finalidade principal é documentar o comportamento de um sistema de maneira clara, concisa e compreensível.]

3 REQUISITOS ESPECIAIS

[Normalmente, um requisito especial é um requisito não-funcional específico de um caso de uso, mas que não é especificado de maneira fácil ou natural no texto do fluxo de eventos do caso de uso. Entre os exemplos de requisitos especiais estão incluídos requisitos legais e reguladores, padrões de aplicativos e atributos de qualidade do sistema a ser criado, incluindo requisitos de usabilidade, confiabilidade, desempenho ou suportabilidade. Além disso, outros requisitos — como ambientes e sistemas operacionais, requisitos de compatibilidade e restrições de design — deverão ser capturados nesta seção.]

3.1 < PRIMEIRO REQUISITO ESPECIAL>

4 PRECONDIÇÕES

[Uma precondição de um caso de uso é o estado do sistema que deve estar presente antes de um caso de uso ser realizado.]

4.1 <PRECONDIÇÃO UM>

5 PÓS-CONDIÇÕES

[Uma pós-condição de um caso de uso é uma lista dos possíveis estados em que o sistema poderá se encontrar imediatamente depois do término de um caso de uso.]

5.1 <PÓS-CONDIÇÃO UM>

6 PONTOS DE EXTENSÃO

[Pontos de extensão do caso de uso.]

6.1 <NOME DO PONTO DE EXTENSÃO>

[Definição da localização do ponto de extensão no fluxo de eventos.]

APÊNDICE C.8 – PROTÓTIPO DA INTERFACE COM O USUÁRIO

<Nome da Empresa>
<Slogan>

<Rua>, Nº <Número> - <Bairro> - <Cidade> - <UF>
CEP: <CEP>

<NOME DO PROJETO>

PROTÓTIPO DA INTERFACE COM O USUÁRIO

VERSÃO 1

Histórico de revisão

Data	Versão	Descrição	Autor

SUMÁRIO

1 PROTÓTIPO DE INTERFACE COM O USUÁRIO 203
1.1 INTRODUÇÃO 203
1.2 FINALIDADE 203
1.3 ESCOPO 203
1.4 REFERÊNCIAS 203
1.5 VISÃO GERAL 203
2 MODELOS 204
2.1 <UM MODELO> 204
2.2 <OUTRO MODELO> 204
2.3 <UM GRUPO DE MODELOS> 204
2.3.1 <Um modelo do grupo> 204
2.3.2 <Outro modelo do grupo> 204

1 PROTÓTIPO DE INTERFACE COM O USUÁRIO

1.1 INTRODUÇÃO

[A introdução do Protótipo de interface com o usuário fornece uma visão geral de todo o documento. Apresente todas as informações que poderão ser necessárias para que o leitor compreenda o documento nesta seção. Este documento é usado para definir a terminologia específica do domínio do problema, explicando termos que podem ser desconhecidos do leitor das descrições de caso de uso ou de outros documentos do projeto. Freqüentemente, este documento poderá ser usado como um dicionário de dados informal, capturando definições de dados para que as descrições de caso de uso e outros documentos do projeto possam concentrar-se no que o sistema deve fazer com as informações. Este documento deverá ser salvo em um arquivo denominado Glossário.]

1.2 FINALIDADE

[Especifique a finalidade deste Protótipo de interface com o usuário.]

1.3 ESCOPO

[Uma breve descrição do escopo deste Protótipo de interface com o usuário; a que Projeto(s) ele está associado e tudo o mais que seja afetado ou influenciado por este documento.]

1.4 REFERÊNCIAS

[Esta subseção fornece uma lista completa de todos os documentos mencionados em qualquer outra parte do Protótipo de interface com o usuário. Identifique cada documento por título, número do relatório (se aplicável), data e organização de publicação. Especifique as fontes a partir das quais as referências podem ser obtidas. Essas informações podem ser fornecidas por um anexo ou outro documento.]

1.5 VISÃO GERAL

[Esta subseção descreve o que o restante do Protótipo de interface com o usuário contém e explica como o documento está organizado.]

2 MODELOS

[Os modelos definidos aqui serão descritos por meio de imagens com o objetivo de facilitar a implementação e apresentação dos modelos de interface de usuário ao cliente.Serão apontados os casos de uso e requisitos relacionados.]

2.1 <UM MODELO>

[O modelo será descrito por meio da imagem e um texto que dará uma breve explicação do que se trata este modelo junto com os requisitos e os modelos de caso de uso relacionados.]

2.2 <OUTRO MODELO>

[O modelo será descrito por meio da imagem e um texto que dará uma breve explicação do que se trata este modelo junto com os requisitos e os modelos de caso de uso relacionados.]

2.3 <UM GRUPO DE MODELOS>

[Conforme necessidade os modelos podem ser agrupado de forma a garantir o melhor entendimento do leitor.]

2.3.1 <Um modelo do grupo>

[O modelo apresentados aqui terá suas características que o identificarão com o grupo que está inserido. Os modelos seguirão o padrão de descrição dos modelos anteriores.]

2.3.2 <Outro modelo do grupo>

[O modelo apresentados aqui terá suas características que o identificarão com o grupo que está inserido. Os modelos seguirão o padrão de descrição dos modelos anteriores.]

APÊNDICE C.9 – MATRIZ DE RASTREABILIDADE

<Nome do projeto>

Histórico de Revisões

Versão:

<Nome da Empresa>

<Slogan>

<Rua>, Nº <Número> - <Bairro> - <Cidade> - <UF>

Data	Versão	Descrição	Autor
[dd/mm/aaaa]	*M.N*	*[Descreva as alterações feitas no documento]*	*[Nome da pessoa que realizou a alteração no documento]*

APÊNDICE C.10 – LISTA DE RISCOS

<Nome do projeto>

Lista de Riscos

Versão:

<Nome da Empres

<Slogan>

<Rua>, Nº <Número> - <Bairro> - <Cidade> - <UF>

Nº	Data	Descrição do Risco	Probabilidade de Ocorrência	Impacto	Status	Responsável
1	*[dd/mm/aaaa]*	*[Descreva aqui o risco observado]*	*[Atla | Média | Baixa]*	*[Atla | Média | Baixa]*	*[Acompanhado | Não-acompanhado | Deixou de existir]*	
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						

APÊNDICE C.11 – PLANO DE PROJETO

<Nome da Empresa>
<Slogan>

<Rua>, Nº <Número> - <Bairro> - <Cidade> - <UF>
CEP: <CEP>

<NOME DO PROJETO>

PLANO DE PROJETO

VERSÃO 1

Histórico de revisão

Data	Versão	Descrição	Autor

SUMÁRIO

1 PLANO DE PROJETO DE SOFTWARE 221
1.1 INTRODUÇÃO 221
1.2 PROPÓSITO 221
1.3 ESCOPO 221
1.4 DEFINIÇÕES, ACRÔNIMOS E ABREVIAÇÕES 221
1.5 REFERÊNCIAS 221
1.6 VISÃO GERAL 221
2 VISÃO DO PROJETO 222
2.1 PROPÓSITO DO PROJETO, ESCOPO E OBJETIVOS 222
2.2 PREMISSAS E RESTRIÇÕES 222
2.3 ARTEFATOS DO PROJETO 222
2.4 EVOLUÇÃO DO PLANO DE DESENVOLVIMENTO DE SOFTWARE 222
3 ORGANIZAÇÃO DO PROJETO 223
3.1 ESTRUTURA ORGANIZACIONAL 223
3.2 CONTATOS EXTERNOS 223
3.3 PAPÉIS E RESPONSABILIDADES 223
4 GERENCIAMENTO DO PROJETO 224
4.1 ESTIMATIVAS DO PROJETO 224
4.2 PLANO DE PROJETO 224
4.2.1 Plano de fases 224
4.2.2 Objetivos das iterações 224
4.2.3 Releases 224
4.2.4 Cronograma do projeto 224
4.2.5 Recursos do projeto 224
4.2.5.1 Plano de staff 224
4.2.5.2 Plano de aquisição de recursos 224
4.2.5.3 Plano de treinamento 225
4.2.6 Orçamento 225
4.3 PLANOS DAS ITERAÇÕES 225
4.4 CONTROLE E ACOMPANHAMENTO DO PROJETO 225
4.4.1 Plano de gerência de requisitos 225

4.4.2 Plano de controle do cronograma 225
4.4.3 Plano de controle do orçamento 225
4.4.4 Plano do controle de qualidade 225
4.4.5 Plano de comunicação 225
4.4.6 Plano de métricas 226
4.5 PLANO DE GERÊNCIA DE RISCOS 226
4.6 PLANO DE ENCERRAMENTO 226
5 PLANOS DE APOIO AO PROCESSO 227
5.1 PLANO DE GERÊNCIA DE CONFIGURAÇÃO 227
5.2 PLANO DE AVALIAÇÃO 227
5.3 PLANO DE DOCUMENTAÇÃO 227
5.4 PLANO DE GARANTIA DE QUALIDADE 227
5.5 PLANO DE SOLUÇÃO DE PROBLEMAS 227
5.6 PLANO DE MELHORIA DO PROCESSO 227

1 PLANO DE PROJETO DE SOFTWARE

1.1 INTRODUÇÃO

[A introdução ao plano deve mostrar uma visão geral de todo o documento. Ele deve incluir o propósito, escopo, definições, acrônimos, abreviações, referencias e uma visão geral deste plano.]

1.2 PROPÓSITO

[Especifique aqui o propósito deste documento.]

1.3 ESCOPO

[Uma breve descrição do escopo; quais as pessoas ou empresas que são afetadas ou influenciadas por este plano.]

1.4 DEFINIÇÕES, ACRÔNIMOS E ABREVIAÇÕES

Consultar o documento Glossário de Negócios.

1.5 REFERÊNCIAS

[Essa sub-seção deve mostrar uma lista completa de todos os documentos referenciados ao longo desse plano. Cada documento deverá ser identificado por título, número do relatório (se aplicável), data e organização de publicação.

1.6 VISÃO GERAL

[Essa sub-seção deve mostrar como o restante desse plano está organizado.]

2 VISÃO DO PROJETO

2.1 PROPÓSITO DO PROJETO, ESCOPO E OBJETIVOS

[Uma breve descrição do propósito e objetivos do projeto.]

2.2 PREMISSAS E RESTRIÇÕES

[Uma lista das premissas do projeto e uma lista das restrições (orçamento, pessoal, equipamentos, cronograma, etc.)]

2.3 ARTEFATOS DO PROJETO

[Uma tabela com os artefatos que serão criados e entregues durante o desenvolvimento do projeto, com as suas respectivas datas de entrega.]

2.4 EVOLUÇÃO DO PLANO DE DESENVOLVIMENTO DE SOFTWARE

[Uma tabela com as possíveis versões desse plano e os critérios para revisões.]

3 ORGANIZAÇÃO DO PROJETO

3.1 ESTRUTURA ORGANIZACIONAL

[Descreve a estrutura organizacional (organograma) do projeto.]

3.2 CONTATOS EXTERNOS

[Descreve como serão os contatos externos ao projeto. Para cada grupo de contato deve ser identificadas as pessoas envolvidas.]

3.3 PAPÉIS E RESPONSABILIDADES

[Identifica os papéis que serão exercidos no projeto e as responsabilidades de cada um.]

4 GERENCIAMENTO DO PROJETO

4.1 ESTIMATIVAS DO PROJETO

[Mostra as estimativas de esforço para o projeto. Pode incluir adicionalmente as estimativas de custo.]

4.2 PLANO DE PROJETO

4.2.1 Plano de fases

[Inclui os seguintes tópicos:

- *Work Breakdown Structure (WBS)*
- *Gráfico de Gantt das atividades*
- *Os principais marcos do projeto e suas datas]*

4.2.2 Objetivos das iterações

[Lista os objetivos que deverão ser alcançados para cada iteração.]

4.2.3 Releases

[Uma breve descrição de cada release.]

4.2.4 Cronograma do projeto

[Cronograma detalhado do projeto.]

4.2.5 Recursos do projeto

4.2.5.1 Plano de staff

[Identifica a quantidade de pessoas e o tipo de papel requerido. Pode ser incrementado com o grau de experiência necessário e a alocação desejada.]

4.2.5.2 Plano de aquisição de recursos

[Descreve como as pessoas deverão ser recrutadas para o projeto.]

4.2.5.3 Plano de treinamento

[Lista dos treinamentos necessários para a execução do projeto e o perfil das pessoas que serão treinadas, além das datas de realização.]

4.2.6 Orçamento

[Alocação de custos de acordo com as atividades da WBS.]

4.3 PLANOS DAS ITERAÇÕES

[Descrição sucinta do planejamento das iterações.]

4.4 CONTROLE E ACOMPANHAMENTO DO PROJETO

4.4.1 Plano de gerência de requisitos

[Descrição sucinta do planejamento de gerência de requisitos.]

4.4.2 Plano de controle do cronograma

[Descrição sobre como o cronograma sera acompanhado e controlado.]

4.4.3 Plano de controle do orçamento

[Descreve como o orçamento sera acompanhado e controlado e as ações corretivas que deverão ser tomadas.]

4.4.4 Plano do controle de qualidade

[Descreve os métodos que serão usados para controlar a qualidade e integridade dos artefatos entregues pelo projeto, além as ações corretivas em cima dos mesmos.]

4.4.5 Plano de comunicação

[Descreve como a comunicação do projeto deve acontecer, quais os documentos de comunicação deverão ser gerado, o formato, a distribuição e a periodicidade.]

4.4.6 Plano de métricas

[Descrição sucinta do plano de coleta, medição e avaliação das métricas.]

4.5 PLANO DE GERÊNCIA DE RISCOS

[Descrição sucinta do planejamento de gerência de riscos.]

4.6 PLANO DE ENCERRAMENTO

[Descrição sucinta sobre as atividades de encerramento do projeto.]

5 PLANOS DE APOIO AO PROCESSO

5.1 PLANO DE GERÊNCIA DE CONFIGURAÇÃO

[Referência para o plano de gerência de configuração]

5.2 PLANO DE AVALIAÇÃO

[Descreve como o produto será homologado, as técnicas de homologação e os critérios de aceitação.]

5.3 PLANO DE DOCUMENTAÇÃO

[Descrição sucinta dos templates utilizados durante o projeto.]

5.4 PLANO DE GARANTIA DE QUALIDADE

[Referência para o plano de garantia de qualidade.]

5.5 PLANO DE SOLUÇÃO DE PROBLEMAS

[Descrição sobre como relatar e solucionar problemas, utilizando as ferramentas de gerência de mudanças.]

5.6 PLANO DE MELHORIA DO PROCESSO

[Descrição sucinta sobre o planejamento para melhorar continuamente o processo de desenvolvimento de software]

APÊNDICE C.12 – DOCUMENTO DE ARQUITETURA DE SOFTWARE

<Nome da Empresa>
<Slogan>

<Rua>, Nº <Número> - <Bairro> - <Cidade> - <UF>
CEP: <CEP>

<NOME DO PROJETO>

DOCUMENTO DE ARQUITETURA DE SOFTWARE

VERSÃO 1

Histórico de revisão

Data	Versão	Descrição	Autor

SUMÁRIO

1 DOCUMENTO DE ARQUITETURA DE SOFTWARE 237
1.1 INTRODUÇÃO 237
1.2 FINALIDADE 237
1.3 ESCOPO 237
1.4 DEFINIÇÕES, ACRÔNIMOS E ABREVIAÇÕES 237
1.5 REFERÊNCIAS 237
1.6 VISÃO GERAL 238
2. REPRESENTAÇÃO ARQUITETURAL 239
3. METAS E RESTRIÇÕES DA ARQUITETURA 240
4. VISÃO DE CASOS DE USO 241
1.7 REALIZAÇÕES DE CASOS DE USO 241
2 VISÃO LÓGICA 242
2.1 VISÃO GERAL 242
2.2 PACOTES DE DESIGN SIGNIFICATIVOS DO PONTO DE VISTA DA ARQUITETURA 242
3 VISÃO DE PROCESSOS 243
4 VISÃO DE IMPLANTAÇÃO 244
5 VISÃO DA IMPLEMENTAÇÃO 245
5.1 VISÃO GERAL 245
5.2 CAMADAS 245
6 VISÃO DE DADOS (OPCIONAL) 246
7 TAMANHO E DESEMPENHO 247
8 QUALIDADE 248

1 DOCUMENTO DE ARQUITETURA DE SOFTWARE

[Este documento oferece uma visão geral arquitetural abrangente do sistema, usando diversas visões arquiteturais para representar diferentes aspectos do sistema. O objetivo deste documento é capturar e comunicar as decisões arquiteturais significativas que foram tomadas em relação ao sistema.]

1.1 INTRODUÇÃO

[A introdução do documento fornece uma visão geral do documento inteiro. Ela inclui a finalidade, o escopo, as definições, os acrônimos, as abreviações, as referências e a visão geral do documento de arquitetura.]

1.2 FINALIDADE

[Esta seção define o papel ou finalidade do Doc. de Arquitetura de Software, na documentação do projeto como um todo, e descreve rapidamente a estrutura do documento. O público-alvo específico do documento é identificado, com uma indicação de como ele espera usar o documento.]

1.3 ESCOPO

[Uma breve descrição da utilidade do Documento de Arquitetura de Software, do que é afetado e/ou influenciado por ele.]

1.4 DEFINIÇÕES, ACRÔNIMOS E ABREVIAÇÕES

[Consultar o documento Glossário de Negócios.]

1.5 REFERÊNCIAS

[Essa sub-seção deve mostrar uma lista completa de todos os documentos referenciados ao longo desse plano. Identifique cada documento por título, número do relatório (se aplicável), e organização de publicação. Especifique as fontes a partir das quais as referências podem ser obtidas].

1.6 VISÃO GERAL

[Descreve o que o restante do documento contém e explica como o documento está organizado.]

2. REPRESENTAÇÃO ARQUITETURAL

[Esta seção descreve qual é a arquitetura de software do sistema atual e como ela é representada. Da visão de casos de uso, visão lógica, visão de processos, visão de implantação e visão de implementação, enumera as visões necessárias e, para cada visão, explica quais tipos de elementos de modelo ela contém.]

3. METAS E RESTRIÇÕES DA ARQUITETURA

[Mostra os requisitos e objetivos do software que têm algum impacto sobre a arquitetura; por exemplo, segurança, garantia, privacidade, uso de um produto desenvolvido internamente e pronto para ser usado, portabilidade, distribuição e reutilização. Ela também captura as restrições especiais que podem ser aplicáveis: estratégia de design e implementação, ferramentas de desenvolvimento, estrutura das equipes, cronograma, código-fonte legado e assim por diante.]

4. VISÃO DE CASOS DE USO

[Esta seção lista casos de uso ou cenários do modelo de casos de uso quando eles representam funcionalidade central e significativa do sistema final ou, quando têm uma grande cobertura arquitetural — eles experimentam muitos elementos arquiteturais ou quando enfatizam ou ilustram um ponto complexo e específico da arquitetura.]

1.7 REALIZAÇÕES DE CASOS DE USO

[Ilustra o funcionamento do software, apresentando algumas realizações (ou cenários) de casos de uso selecionadas e explica como os diversos elementos do modelo de design contribuem para a respectiva funcionalidade.]

2 VISÃO LÓGICA

[Esta seção descreve as partes significativas do ponto de vista da arquitetura do modelo de design, como sua divisão em subsistemas e pacotes. Para cada pacote significativo, ela mostra sua divisão em classes e utilitários de classe, apresentando as classes significativas do ponto de vista da arquitetura e descrevendo as suas responsabilidades, bem como alguns relacionamentos, operações e atributos de grande importância.]

2.1 VISÃO GERAL

[Esta subseção descreve toda a decomposição do modelo de design em termos de camadas e de hierarquia de pacotes.]

2.2 PACOTES DE DESIGN SIGNIFICATIVOS DO PONTO DE VISTA DA ARQUITETURA

[Para cada pacote significativo, inclua uma subseção com o respectivo nome, uma breve descrição e um diagrama com todos os pacotes e classes significativos nele contidos. Para cada classe significativa no pacote, inclua o respectivo nome, uma breve descrição e, opcionalmente, uma descrição de algumas das suas principais responsabilidades, operações e atributos.]

3 VISÃO DE PROCESSOS

[Descreve a decomposição do sistema em processos leves (threads simples de controle) e processos pesados (agrupamentos de processos leves). Organize a seção em grupos de processos que se comunicam ou interagem. Descreva os modos principais de comunicação entre processos, como transmissão de mensagens e interrupções.]

4 VISÃO DE IMPLANTAÇÃO

[Esta seção descreve uma ou mais configurações da rede física (hardware) na qual o software é implantado e executado. Ela é uma visão do Modelo de Implantação. No mínimo, para cada configuração, ela deve indicar os nós físicos (computadores, CPUs) que executam o software e suas interconexões (barramento, LAN, ponto a ponto, etc.)

5 VISÃO DA IMPLEMENTAÇÃO

[Descrever a estrutura geral do modelo de implementação, a divisão do software em camadas e os subsistemas no modelo de implementação e todos os componentes significativos do ponto de vista da arquitetura.]

5.1 VISÃO GERAL

[Esta subseção nomeia e define as diversas camadas e o seu conteúdo, as regras que determinam a inclusão em uma camada específica e as fronteiras entre as camadas. Inclua um diagrama de componentes que mostre os relacionamentos entre as camadas.]

5.2 CAMADAS

[Para cada camada, inclua uma subseção com o respectivo nome, uma lista dos subsistemas localizados na camada e um diagrama de componentes.]

6 VISÃO DE DADOS (OPCIONAL)

[Descrição da perspectiva de armazenamento de dados persistentes do sistema. Esta seção será opcional se os dados persistentes forem poucos ou inexistentes ou se a conversão entre o modelo de design e o modelo de dados for trivial.]

7 TAMANHO E DESEMPENHO

[Principais características de dimensionamento do software que têm um impacto na arquitetura, bem como as restrições do desempenho desejado.]

8 QUALIDADE

[Uma descrição de como a arquitetura do software contribui para todos os recursos (exceto a funcionalidade) do sistema: extensibilidade, confiabilidade, portabilidade e assim por diante. Se essas características possuírem significado especial, como implicações de segurança garantiam ou privacidade, elas deverão ser delineadas claramente.]

APÊNDICE C.13 – MODELO DE IMPLEMENTAÇÃO

<Nome da Empresa>
<Slogan>

<Rua>, Nº <Número> - <Bairro> - <Cidade> - <UF>
CEP: <CEP>

<NOME DO PROJETO>

MODELO DE IMPLEMENTAÇÃO DE SOFTWARE

VERSÃO 1

Histórico de revisão

Data	Versão	Descrição	Autor

SUMÁRIO

1 MODELO DE IMPLEMENTAÇÃO DE SOFTWARE 257
1.1 INTRODUÇÃO 257
1.2 FINALIDADE 257
1.3 ESCOPO 257
1.4 DEFINIÇÕES, ACRÔNIMOS E ABREVIAÇÕES 257
1.5 REFERÊNCIAS 257
1.6 VISÃO GERAL 258
2 PADROES DE CONSTRUÇÃO 259
2.1 MODELO ESTRUTURAL 259
2.2 TERMINOLOGIA 259
2.3 DOCUMENTAÇÃO 259

1 MODELO DE IMPLEMENTAÇÃO DE SOFTWARE

1.1 INTRODUÇÃO

[A finalidade deste documento é coletar, analisar e definir as necessidades e características de nível superior do <<Nome do Sistema>>. Ele se concentra nos recursos necessários aos envolvidos e aos usuários-alvo e nas razões que levam a essas necessidades. Os detalhes de como o <<Nome do Sistema>> atende a essas necessidades estão descritos nas especificações suplementares e de caso de uso.]

[A introdução do documento de Visão oferece uma visão geral de todo o documento. Ela contém a finalidade, o escopo, as definições, os acrônimos, as abreviações, as referências e a visão geral deste documento Visão.]

1.2 FINALIDADE

[Especifique a finalidade deste documento Visão.]

1.3 ESCOPO

[Uma breve descrição do escopo deste documento Visão; a que Projeto(s) ele está associado e tudo o mais que seja afetado ou influenciado por este documento.]

1.4 DEFINIÇÕES, ACRÔNIMOS E ABREVIAÇÕES

[Estas informações estão presentes no glossário.]

1.5 REFERÊNCIAS

[Esta subseção apresenta uma lista completa de todos os documentos mencionados no documento de Visão. Identifique cada documento por título, número do relatório (se aplicável), data e organização de publicação. Especifique as fontes a partir das quais as referências podem ser obtidas. Essas informações podem ser fornecidas por um anexo ou outro documento.]

1.6 VISÃO GERAL

[Esta subseção descreve o que o restante do documento Visão contém e explica como o documento está organizado.]

2 PADROES DE CONSTRUÇÃO

2.1 MODELO ESTRUTURAL

[Forneça o padrão da estrutura visual das classes a serem criadas . Será definido onde serão descritos comentários, instanciação de objetos, endentação e etc. De modo geral será mostrado o modelo de classe/função da empresa.]

2.2 TERMINOLOGIA

[Nessa seção serão definidos como serão nomeadas os elementos estruturais que constituem o projeto segundo necessidade e paradigmas usados.

- *Variáveis – Forneça a regra de como serão atribuídos os seus nomes*
- *Métodos – Forneça a regra de como serão atribuídos os seus nomes*
- *Atributos – Forneça a regra de como serão atribuídos os seus nomes*
- *Classes – Forneça a regra de como serão atribuídos os seus nomes*
- *Pacotes - Forneça a regra de como serão atribuídos os seus nomes*
- *Funções - Forneça a regra de como serão atribuídos os seus nomes*
- *Sub-funções - Forneça a regra de como serão atribuídos os seus nomes]*

2.3 DOCUMENTAÇÃO

[Forneça um documentação do código fonte.Defina como o autor organizará as informações autorais e de versão no código]

APÊNDICE C.14 – PLANO DE INTEGRAÇÃO DO BUILD

<Nome da Empresa>
<Slogan>

<Rua>, Nº <Número> - <Bairro> - <Cidade> - <UF>
CEP: <CEP>

<NOME DO PROJETO>

PLANO DE INTEGRAÇÃO DO BUILD

VERSÃO 1

Histórico de revisão

Data	Versão	Descrição	Autor

SUMÁRIO

1 PLANO DE INTEGRAÇÃO DO BUILD 269
1.1 INTRODUÇÃO 269
1.2 FINALIDADE 269
1.3 ESCOPO 269
1.4 DEFINIÇÕES, ACRÔNIMOS E ABREVIAÇÕES 269
1.5 REFERÊNCIAS 269
1.6 VISÃO GERAL 269
2 SUBSISTEMAS 270
3 BUILDS 271

1 PLANO DE INTEGRAÇÃO DO BUILD

1.1 INTRODUÇÃO

[A introdução do Plano de integração do build oferece uma visão geral de todo o documento. Ela inclui a finalidade, o escopo, as definições, os acrônimos, as abreviações, as referências e uma visão geral deste Plano de integração do build.]

1.2 FINALIDADE

[Especifique a finalidade deste Plano de integração do build.]

1.3 ESCOPO

[Uma breve descrição do escopo deste Plano de integração do build; os modelos aos quais ele está associado e tudo o que é afetado ou influenciado por este documento.]

1.4 DEFINIÇÕES, ACRÔNIMOS E ABREVIAÇÕES

[Consultar o documento de Glossário de negócios]

1.5 REFERÊNCIAS

[Esta subseção apresenta uma lista completa de todos os documentos mencionados no Plano de integração do build. Identifique cada documento por título, número do relatório (se aplicável), data e organização de publicação. Especifique as fontes a partir das quais as referências podem ser obtidas. Essas informações podem ser fornecidas por um anexo ou outro documento.]

1.6 VISÃO GERAL

[Esta subseção descreve o conteúdo restante do Plano de integração do build e explica como o documento está organizado.]

2 SUBSISTEMAS

[Especifique os subsistemas que deverão ser implementados nesta iteração. Especifique também a ordem preferencial na qual os subsistemas serão implementados para que estejam prontos no devido tempo para a integração.]

3 BUILDS

[A integração, na iteração, está dividida em uma série de incrementos, cada um resultando em um build, que é testado para verificar a integração. Esta seção especifica os builds que devem ser criados e os subsistemas que farão parte de cada build. Para cada build, esta seção deve especificar como ele é construído, os critérios para sua avaliação e como ele deverá ser testado, especificamente:

- *Construção*
    - *Scripts de build e quaisquer outras instruções que descrevem como o build será construído.*
    - *Registros de baseline que definem as versões dos itens de configuração usados para construir o build.*
- *Avaliação e Teste*
    - *Critérios de avaliação — uma descrição dos recursos em relação aos quais o build será avaliado. Esses critérios poderão incluir um subconjunto dos critérios de avaliação no Plano de Iteração correspondente e outros critérios de avaliação específicos do build (especialmente quando, por exemplo, tratar-se de um build de arquitetura que não ofereça muitos ou talvez nenhum recurso que seja visível para o usuário final.*
    - *Instruções de instalação e configuração para executar e testar o build.*
    - *Casos de teste, procedimentos de teste, scripts de teste e resultados de teste.*

Observação: Em todos os casos, não é necessário replicar o material neste plano — as referências serão suficientes se o material estiver presente em outros artefatos; por exemplo, no Artefato Plano de Teste de Iteração.]

APÊNDICE C.15 – PLANO DE TESTE

<Nome da Empresa>
<Slogan>

<Rua>, Nº <Número> - <Bairro> - <Cidade> - <UF>
CEP: <CEP>

<NOME DO PROJETO>

PLANO DE TESTES

VERSÃO 1

Histórico de revisão

Data	Versão	Descrição	Autor

SUMÁRIO

1 PLANO DE TESTE 281
1.1 INTRODUÇÃO 281
1.2 FINALIDADE 281
1.3 ESCOPO 281
1.4 PÚBLICO-ALVO 281
1.5 TERMINOLOGIA E ACRÔNIMOS DO DOCUMENTO 282
1.6 REFERÊNCIAS 282
2 MISSÃO DE AVALIAÇÃO E MOTIVAÇÃO DOS TESTES 283
2.1 MISSÃO DE AVALIAÇÃO 283
3 ITENS DE TESTE-ALVO 284
4 RESUMO DOS TESTES PLANEJADOS 285
4.1 RESUMO DAS INCLUSÕES DOS TESTES 285
4.2 RESUMO DAS EXCLUSÕES DOS TESTES 285
5 ABORDAGEM DOS TESTES 286
5.1 TIPOS E TÉCNICAS DE TESTE 286
5.1.1 Teste de Integridade de Dados e de Banco de Dados 286
OBJETIVO DA TÉCNICA: 286
TÉCNICA: 286
ESTRATÉGIAS: 286
FERRAMENTAS NECESSÁRIAS: 286
CONSIDERAÇÕES ESPECIAIS: 287
5.1.2 Teste de Funcionamento 287
5.1.3 Teste de Interface do Usuário 287
5.1.4 Teste de Segurança e de Controle de Acesso 288
6 CRITÉRIOS DE ENTRADA E DE SAÍDA 289
6.1 PLANO DE TESTE 289
6.1.1 Critérios de Entrada de Plano de Teste 289
6.1.2 Critérios de Saída de Plano de Teste 289
6.1.3 Critérios de Suspensão e de Reinício 289
6.2 CICLOS DE TESTE 290
6.2.1 Critérios de Entrada de Ciclo de Teste 290

6.2.2 Critérios de Saída de Ciclo de Teste 290
6.2.3 Término Anormal do Ciclo de Teste 290
7 PRODUTOS LIBERADOS 291
7.1 SUMÁRIOS DE AVALIAÇÃO DE TESTES 291
7.1.1 Resultados Detalhados dos Testes 291
7.1.2 Matrizes de Rastreabilidade 291
8 FLUXO DE TRABALHO DE TESTE 292
9 NECESSIDADES AMBIENTAIS 293
9.1 HARDWARE BÁSICO DO SISTEMA 293
9.2 ELEMENTOS DE SOFTWARE BÁSICOS DO AMBIENTE DE TESTE 293
9.3 FERRAMENTAS DE PRODUTIVIDADE E DE SUPORTE 294
9.4 CONFIGURAÇÕES DO AMBIENTE DE TESTE 294
10 MARCOS DA ITERAÇÃO 296
11 PROCEDIMENTOS E PROCESSOS DE GERENCIAMENTO 297
11.1 MEDIÇÃO E AVALIAÇÃO DA EXTENSÃO DO TESTE 297
11.2 GERAÇÃO DE RELATÓRIOS SOBRE COBERTURA DE TESTE 297
11.3 APROVAÇÃO E ENCERRAMENTO 297

1 PLANO DE TESTE

1.1 INTRODUÇÃO

1.2 FINALIDADE

[A finalidade do Plano de Teste de Iteração é reunir todas as informações necessárias para planejar e controlar o esforço de teste referente a uma iteração específica. Ele descreve a abordagem dada ao teste do software e é o plano de nível superior gerado e usado pelos gerentes para coordenar o esforço de teste. Assim ele:

- *Identifica os itens que devem ser inspecionados pelos testes;*
- *Identifica a motivação e as idéias subjacentes às áreas de teste a serem abrangidas;*
- *Descreve a abordagem de teste que será usada;*
- *Identifica os recursos necessários e fornece uma estimativa dos esforços de teste;*
- *Lista os elementos liberados do projeto de teste.]*

1.3 ESCOPO

[Descreva os níveis de teste (por exemplo, Unidade, Integração ou Sistema, e os tipos de teste (como, por exemplo, Funcionalidade, Usabilidade, Confiabilidade, Desempenho e Suportabilidade) que serão abordados por este Plano de Teste. Também é importante fornecer uma indicação geral das áreas importantes que serão excluídas do escopo, especialmente nos casos em que o público-alvo possa supor sensatamente que elas serão incluídas.
Observação: Evite incluir detalhes aqui que serão repetidos nas seções Itens de Teste-Alvo e Resumo dos Testes Planejados.]

1.4 PÚBLICO-ALVO

[Forneça uma breve descrição do público para o qual o Plano de Teste está sendo escrito. Isso ajudará os leitores do documento a identificarem se ele realmente está destinado ao seu uso e também ajudará a evitar que o documento seja usado de forma inadequada.
Observação: Freqüentemente o estilo e o conteúdo do documento são alterados em função do público-alvo.]

1.5 TERMINOLOGIA E ACRÔNIMOS DO DOCUMENTO

[Essa informação estará presente no glosário.]

1.6 REFERÊNCIAS

[Esta subseção fornece uma lista dos documentos mencionados em qualquer outra parte do Plano de Teste. Identifique cada documento por título, número da versão (ou do relatório, se aplicável), data, organização de publicação ou autor original. Evite listar documentos que exercem influência no contexto mas que não foram mencionados diretamente. Especifique as fontes a partir das quais as "versões oficiais" das referências podem ser obtidas como, por exemplo, nomes UNC de intranet ou códigos de referência de documento. Essas informações podem ser fornecidas por um anexo ou outro documento.]

2 MISSÃO DE AVALIAÇÃO E MOTIVAÇÃO DOS TESTES

[Forneça uma visão geral da missão e da motivação dos testes que serão conduzidos nesta iteração.]

2.1 MISSÃO DE AVALIAÇÃO

[Forneça uma breve sentença que defina a missão do esforço de avaliação na iteração atual. Essa sentença poderá incorporar uma ou mais preocupações incluindo:

- *Localizar o maior número de erros possível;*
- *Localizar problemas importantes;*
- *Avaliar os riscos da qualidade perceptível;*
- *Informar sobre os riscos perceptíveis do projeto;*
- *Certificar segundo um padrão;*
- *Verificar uma especificação (requisitos, design ou alegações);*
- *Informar sobre a qualidade do produto;*
- *Satisfazer os envolvidos;*
- *Informar sobre os testes;*
- *Cumprir as determinações do processo;*
- *E assim por diante.*

Cada missão fornece um contexto diferente para o esforço de teste e altera a maneira como o teste deverá ser abordado.]

3 ITENS DE TESTE-ALVO

[A listagem abaixo identifica os itens de software, de hardware e elementos de suporte do produto que foram identificados como objetivos dos testes. Esta lista representa os itens que serão testados.
Forneça uma lista de nível superior dos principais itens que estarão sujeitos a teste. Essa lista deve incluir itens produzidos diretamente pela equipe de desenvolvimento do projeto e itens de que dependem esses produtos. Por exemplo, o hardware de processamento básico, dispositivos periféricos, sistemas operacionais, produtos ou componentes de terceiros etc. É recomendável agrupar a lista por categoria e atribuir importância relativa a cada motivador.]

4 RESUMO DOS TESTES PLANEJADOS

[Esta seção apresenta os recursos recomendados para o projeto <Nome do Projeto>, suas principais responsabilidades e seu conjunto de conhecimentos ou de habilidades.]

4.1 RESUMO DAS INCLUSÕES DOS TESTES

[Esta seção fornece um resumo de nível superior dos testes que serão executados. O resumo fornecido aqui representa uma visão geral de nível superior dos testes que serão e dos que não serão executados.]

4.2 RESUMO DAS EXCLUSÕES DOS TESTES

[Forneça um resumo de nível superior dos possíveis testes que poderiam ter sido conduzidos, mas que foram explicitamente excluídos deste plano. Se você não for implementar ou executar um tipo de teste, informe claramente que o teste não será executado ou implementado e justifique por que. A seguir, há exemplos de justificativas que poderão ser usadas:

- *Esses testes não contribuem para alcançar a missão de avaliação.*
- *Não há recursos suficientes para executar esses testes.*
- *Esses testes são desnecessários devido aos testes executados por xxxx.*

Segundo um prisma heurístico, se você achar que é perfeitamente concebível que um dos membros de seu público espere que um determinado aspecto de teste seja incluído e se você não pretender ou não puder incluí-lo, justifique sua exclusão. Se a equipe concordar que a exclusão é óbvia, você provavelmente não precisará listá-la.]

5 ABORDAGEM DOS TESTES

[Esta seção apresenta a estratégia recomendada para criar e implementar os testes necessários. As seções 3, Itens de Teste-Alvo, e 4, Resumo dos Testes Planejados, identificaram que itens serão testados e que tipos de testes serão executados. Esta seção descreve como esses testes serão realizados.
Um aspecto a ser considerado na abordagem dos testes é as técnicas que serão usadas. Deverá ser incluído um resumo de como cada técnica poderá ser implementada, de uma perspectiva manual e/ou automatizada, e os critérios para comprovar que a técnica é útil e eficaz. Para cada técnica, forneça uma descrição a seu respeito e defina por que é uma parte importante da abordagem dos testes resumindo brevemente como ela ajuda a alcançar a Missão de Avaliação ou como aborda os Motivadores dos Testes.
Outro aspecto a ser discutido nesta seção é os modelos de Erro ou Falha que são aplicáveis e as maneiras de abordar como avaliá-los.]

5.1 TIPOS E TÉCNICAS DE TESTE

5.1.1 Teste de Integridade de Dados e de Banco de Dados

[Os bancos de dados e os processos de banco de dados deverão ser testados como um subsistema independente. Esse teste deve testar os subsistemas sem que a Interface do Usuário do objetivo do teste faça interface com os dados.]

Objetivo da técnica:	*[Experimentar processos e métodos de acesso a banco de dados independentes da UI para que você possa observar e registrar comportamentos-alvo incorretos ou a existência de dados corrompidos.]*
Técnica:	*[Dispare cada processo e método de acesso a banco de dados, propagando dados válidos e inválidos ou solicitações de dados em cada um deles.* *Inspecione o banco de dados para assegurar que os dados foram distribuídos conforme o planejado e que todos os eventos de banco de dados ocorreram de forma adequada, ou revise os dados retornados para assegurar que os dados corretos foram recuperados pelas razões corretas.]*
Estratégias:	*[Descreva uma ou mais estratégias que podem ser usadas pela técnica para observar, de forma precisa, os resultados do teste. A estratégia combina elementos do método através do qual a observação pode ser feita e das características dos resultados específicos que indicam um provável êxito ou falha do teste. O ideal é que as estratégias sejam autoverificadas, permitindo que os testes automatizados façam uma avaliação inicial do êxito ou falha do teste. No entanto, tenha atenção para reduzir os riscos inerentes à determinação automática dos resultados.]*
Ferramentas necessárias:	*[A técnica exige as seguintes ferramentas:* • *Ferramenta de Automação de Scripts de Teste;* • *Restaurador e reprodutor de imagem da configuração básica;* • *Ferramentas de backup e de recuperação;* • *Ferramentas de monitoramento de instalação (registro, disco rígido, CPU, memória etc.);* • *Ferramentas e utilitários SQL de banco de dados;* • *Ferramentas de geração de dados.]*

Considerações especiais:	• *[Os testes poderão necessitar de drivers ou de um ambiente de desenvolvimento SGBD para inserir ou modificar dados diretamente no banco de dados.* • *Os processos deverão ser disparados manualmente.* • *Deverão ser usados bancos de dados pequenos ou de tamanho mínimo (com um número limitado de registros) para aumentar a visibilidade de quaisquer eventos não aceitáveis.]*

5.1.2 Teste de Funcionamento

[O teste de funcionamento do objetivo do teste deve concentrar-se em todos os requisitos de teste que possam ser diretamente associados a casos de uso ou funções e regras de negócios. Esse teste tem por fim verificar a adequada aceitação, processamento e recuperação dos dados, e a implementação apropriada das regras de negócios. Esse tipo de teste baseia-se em técnicas de caixa preta; ou seja, verificar o aplicativo e seus processos internos interagindo com o aplicativo através da Interface Gráfica do Usuário (GUI) e analisar a saída ou os resultados. A tabela a seguir identifica um resumo do teste recomendado para cada aplicativo.]

Objetivo da Técnica:	*[Experimentar a funcionalidade do objetivo do teste, incluindo a navegação, a entrada, o processamento e a recuperação de dados a fim de observar e registrar o comportamento-alvo.]*
Técnica:	*[Experimentar os recursos e fluxos ou funções de cada um dos cenários de caso de uso, utilizando dados válidos e inválidos para verificar se:* • *Os resultados esperados ocorrerão quando forem usados dados válidos;* • *As mensagens de erro ou de aviso apropriadas serão exibidas quando forem usados dados inválidos;* • *Cada regra de negócio será aplicada de forma adequada.]*
Estratégias:	*[Descreva uma ou mais estratégias que podem ser usadas pela técnica para observar, de forma precisa, os resultados do teste. A estratégia combina elementos do método através do qual a observação pode ser feita e das características dos resultados específicos que indicam um provável êxito ou falha do teste. O ideal é que as estratégias sejam autoverificadas, permitindo que os testes automatizados façam uma avaliação inicial do êxito ou falha do teste. No entanto, tenha atenção para reduzir os riscos inerentes à determinação automática dos resultados.]*
Ferramentas Necessárias:	*[A técnica exige as seguintes ferramentas:* • *Ferramenta de Automação de Scripts de Teste;* • *Restaurador e reprodutor de imagem da configuração básica;* • *Ferramentas de backup e de recuperação;* • *Ferramentas de monitoramento de instalação (registro, disco rígido, CPU, memória etc.);* • *Ferramentas de geração de dados.]*
Critérios de Êxito:	*[A técnica suporta o teste de:* • *Todos os principais cenários de caso de uso;* • *Todos os principais recursos.*
Considerações Especiais:	*[Identifique ou descreva os itens ou problemas (internos ou externos) que exercem influência sobre a implementação e a execução do teste de funcionamento.]*

5.1.3 Teste de Interface do Usuário

[O Teste de Interface do Usuário (UI) verifica a interação do usuário com o software. O teste de UI tem por fim assegurar que a UI forneça ao usuário o acesso e a navegação adequados através das funções do objetivo do teste. Além disso, o teste de UI assegura que os objetos

contidos na UI funcionem conforme o esperado e estejam em conformidade com padrões corporativos ou da indústria.]

Objetivo da Técnica:	*[Experimentar o seguinte para observar e registrar a conformidade com padrões e o comportamento-alvo:* • *A navegação pelo objetivo do teste para verificar se reflete os requisitos e funções de negócios, incluindo a navegação janela a janela e campo a campo, e o uso de métodos de acesso (teclas de tabulação, movimentos do mouse e teclas aceleradoras).* • *As características e os objetos das janelas poderão ser experimentados como, por exemplo, menus, tamanho, posição, estado e foco.]*
Técnica:	*[Crie ou modifique testes para cada janela a fim de verificar a navegação adequada e os estados de objeto apropriados para cada janela e objeto do aplicativo.]*
Estratégias:	*[Descreva uma ou mais estratégias que podem ser usadas pela técnica para observar, de forma precisa, os resultados do teste. A estratégia combina elementos do método através do qual a observação pode ser feita e das características dos resultados específicos que indicam um provável êxito ou falha do teste. O ideal é que as estratégias sejam autoverificadas, permitindo que os testes automatizados façam uma avaliação inicial do êxito ou falha do teste. No entanto, tenha atenção para reduzir os riscos inerentes à determinação automática dos resultados.]*
Ferramentas Necessárias:	*[A técnica necessita da Ferramenta de Automação de Scripts de Teste.]*
Critérios de Êxito:	*[A técnica suporta o teste de cada tela ou janela principal que será muito usada pelo usuário final.]*
Considerações Especiais:	*[Nem todas as propriedades referentes a objetos personalizados e de terceiros poderão ser acessadas.]*

5.1.4 Teste de Segurança e de Controle de Acesso

[O Teste de Segurança e de Controle de Acesso concentra-se em duas áreas de segurança principais:
Segurança no nível do aplicativo, incluindo o acesso aos Dados ou às Funções de Negócios
Segurança no nível do sistema, incluindo efetuar login ou acessar remotamente o sistema
Com base no nível de segurança desejado, a segurança no nível do aplicativo assegura que os atores estejam restritos a funções ou casos de uso específicos, ou que tenham acesso limitado aos dados disponíveis. Por exemplo, todos têm permissão para inserir dados e criar novas contas, mas apenas os gerentes poderão excluí-los. Se houver segurança no nível dos dados, o teste assegurará que o "tipo de usuário um" possa ver todas as informações de um cliente, incluindo dados financeiros. No entanto, o "tipo de usuário dois" somente verá os dados demográficos referentes ao mesmo cliente.
A segurança no nível do sistema assegura que somente os usuários a que tenha sido concedido acesso ao sistema serão capazes de acessar os aplicativos e somente através dos gateways apropriados.]

Objetivo da Técnica:	*[Experimentar o objetivo do teste nas seguintes condições para observar e registrar o comportamento-alvo:* • *Segurança no nível do aplicativo: um ator poderá acessar somente as funções ou os dados para o quais seu tipo de usuário tenha recebido permissão;* • *Segurança no nível do sistema: somente os atores com acesso ao sistema e aos aplicativos têm permissão para acessá-los].*
Técnica:	*[Segurança no nível do aplicativo: Identifique e liste cada tipo de usuário e as funções ou os*

	dados para os quais cada tipo tem permissão de acesso. • *Crie testes para cada tipo de usuário e verifique cada permissão criando transações específicas para cada tipo de usuário;* • *Modifique o tipo de usuário e execute novamente os testes para os mesmos usuários. Em cada caso, verifique se as funções ou dados adicionais estão corretamente disponíveis ou se têm seu acesso negado.]*
Estratégias:	*[Descreva uma ou mais estratégias que podem ser usadas pela técnica para observar, de forma precisa, os resultados do teste. A estratégia combina elementos do método através do qual a observação pode ser feita e das características dos resultados específicos que indicam um provável êxito ou falha do teste. O ideal é que as estratégias sejam autoverificadas, permitindo que os testes automatizados façam uma avaliação inicial do êxito ou falha do teste. No entanto, tenha atenção para reduzir os riscos inerentes à determinação automática dos resultados.]*
Ferramentas Necessárias:	*[A técnica exige as seguintes ferramentas:* • *Ferramenta de Automação de Scripts de Teste;* • *Ferramentas de investigação e contra a violação da segurança por "hackers";* • *Ferramentas de Administração da Segurança do Sistema Operacional]*
Critérios de Êxito:	*[A técnica suporta o teste das funções apropriadas. É possível também que os dados afetados pelas configurações de segurança sejam testados para cada tipo de ator conhecido.]*
Considerações Especiais:	*[O acesso ao sistema deverá ser revisto ou discutido com o administrador de sistemas ou de rede adequado. Talvez esse teste não seja necessário, já que poderá ser uma das funções da administração de sistemas ou de rede.]*

6 CRITÉRIOS DE ENTRADA E DE SAÍDA

6.1 PLANO DE TESTE

6.1.1 Critérios de Entrada de Plano de Teste

[Especifique os critérios que serão usados para determinar se a execução do Plano de Teste poderá ser iniciada.]

6.1.2 Critérios de Saída de Plano de Teste

[Especifique os critérios que serão usados para determinar se a execução do Plano de Teste foi concluída ou se a continuação da execução não será vantajosa.]

6.1.3 Critérios de Suspensão e de Reinício

[Especifique os critérios que serão usados para determinar se os testes deverão ser prematuramente suspensos ou concluídos antes que o plano tenha sido totalmente executado. Especifique também segundo que critérios os testes poderão ser reiniciados.]

6.2 CICLOS DE TESTE

6.2.1 Critérios de Entrada de Ciclo de Teste

[Especifique os critérios que serão usados para determinar se o esforço do próximo Ciclo de Teste deste Plano de Teste poderá ser iniciado.]

6.2.2 Critérios de Saída de Ciclo de Teste

[Especifique os critérios que serão usados para determinar se o esforço de teste do Ciclo de Teste atual deste Plano de Teste é considerado suficiente.]

6.2.3 Término Anormal do Ciclo de Teste

[Especifique os critérios que serão usados para determinar se os testes deverão ser prematuramente suspensos ou concluídos para o ciclo de teste atual, ou se o futuro build a ser testado deverá ser alterado.]

7 PRODUTOS LIBERADOS

[Nesta seção, liste os vários artefatos que serão criados pelo esforço de teste e que serão produtos liberados úteis aos vários envolvidos do esforço de teste. Não liste todos os produtos do trabalho; liste apenas os que propiciam benefícios diretos tangíveis aos envolvidos e os que permitem medir o êxito do esforço de teste.]

7.1 SUMÁRIOS DE AVALIAÇÃO DE TESTES

[Forneça um breve resumo da forma e do conteúdo dos sumários de avaliação de testes e indique com que freqüência eles serão gerados.]

7.1.1 Resultados Detalhados dos Testes

[Trata-se de um conjunto de planilhas do Microsoft Excel relacionando os resultados determinados para cada caso de teste ou refere-se ao repositório dos registros de testes e dos resultados determinados mantidos por um produto de teste especializado.]

7.1.2 Matrizes de Rastreabilidade

[Utilizando uma ferramenta como o Rational RequisistePro ou o Microsoft Excel, forneça uma ou mais matrizes de relacionamentos de rastreabilidade entre os itens rastreados.]

8 FLUXO DE TRABALHO DE TESTE

[Forneça um resumo do fluxo de trabalho a ser seguido pela equipe de teste no desenvolvimento e na execução deste Plano de Teste.
fluxo de trabalho de teste específico que você usará deve ser documentado separadamente no Caso de Desenvolvimento do projeto. Ele deve explicar como o projeto personalizou o fluxo de trabalho de teste básico do RUP (normalmente fase a fase). Na maior parte dos casos, é recomendável que, nesta seção do Plano de Teste, você insira uma referência à seção relevante do Caso de Desenvolvimento. Poderá ser útil e suficiente simplesmente incluir um diagrama ou uma imagem ilustrando o fluxo de trabalho de teste.

Os detalhes mais específicos das tarefas de teste individuais poderão ser definidos de várias maneiras diferentes, dependendo da cultura do projeto. Veja os exemplos a seguir:
poderão ser definidos como uma lista de tarefas nesta seção do Plano de Teste ou em um apêndice complementar
poderão ser definidos em uma programação central do projeto (freqüentemente em uma ferramenta de programação como o Microsoft Project)
poderão ser documentados em listas de tarefas "dinâmicas" individuais para cada membro da equipe, que geralmente são muito detalhadas para serem inseridas no Plano de Teste
poderão ser documentados em um quadro branco localizado em um local central e atualizado dinamicamente
poderão simplesmente não serem documentados formalmente
Com base na cultura de seu projeto, você deverá listar suas tarefas de teste específicas aqui ou fornecer um texto descritivo explicando o processo utilizado por sua equipe para efetuar o planejamento detalhado de tarefas. Você também poderá fazer referência ao local em que os detalhes serão armazenados, se for adequado.
Para os Planos de Teste Mestre, é recomendável evitar o planejamento detalhado de tarefas, que freqüentemente será um esforço improdutivo se efetuado como uma atividade antecipada no início do projeto. Um Plano de Teste Mestre poderá descrever, de maneira útil, as fases e o número de iterações, e fornecer uma indicação dos tipos de teste que geralmente são planejados para cada Fase ou Iteração.
Observação: Nos casos em que as informações referentes a processos e ao planejamento detalhado forem registradas em um local central e separadamente deste Plano de Teste, você terá que gerenciar os problemas originados pelo fato de existirem cópias duplicadas das mesmas informações. Para evitar que os membros da equipe consultem informações desatualizadas, é recomendável, nesse caso, inserir o mínimo possível de informações sobre processos e planejamento no Plano de Teste para facilitar a constante manutenção das informações e, portanto, simplesmente fazer referência ao material que se encontra no "Plano Mestre".]

9 NECESSIDADES AMBIENTAIS

[Esta seção apresenta os recursos não humanos necessários ao Plano de Teste.]

9.1 HARDWARE BÁSICO DO SISTEMA

Os conjuntos de tabelas a seguir apresentam os recursos do sistema necessários ao esforço de teste descrito neste Plano de Teste.
[É possível que os elementos específicos do sistema de teste não sejam totalmente compreendidos nas iterações iniciais, sendo assim, espera-se que esta seção seja preenchida ao logo do tempo. É recomendável que o sistema simule o ambiente de produção, reduzindo o acesso concorrente e o tamanho do banco de dados, se e quando for necessário. Observação: Adicione ou exclua itens conforme o necessário.]

Recurso	Quantidade	Nome e Tipo
Servidor de Banco de Dados		
Rede ou Sub-rede		
Nome do Servidor		
Nome do Banco de Dados		
PCs de Teste Cliente		
Inclua requisitos de configuração especiais		
Repositório de Teste		
Rede ou Sub-rede		
Nome do Servidor		
PCs de Desenvolvimento de Teste		

9.2 ELEMENTOS DE SOFTWARE BÁSICOS DO AMBIENTE DE TESTE

São necessários os seguintes elementos de software básicos no ambiente de teste deste Plano de Teste.
[Observação: Adicione ou exclua itens conforme o necessário.]

Nome do Elemento de Software	Versão	Tipo e Outras Observações
NT Workstation		Sistema Operacional
Windows 2000		Sistema Operacional
Internet Explorer		Navegador da Internet

Netscape Navigator		Navegador da Internet
Microsoft Outlook		Software Cliente de E-Mail
Network Associates McAffee Virus Checker		Software de Detecção e Recuperação de Vírus

9.3 FERRAMENTAS DE PRODUTIVIDADE E DE SUPORTE

Serão utilizadas as seguintes ferramentas para suportar o processo de teste deste Plano de Teste.
[Observação: Adicione ou exclua itens conforme o necessário.]

Categoria ou Tipo de Ferramenta	Nome da Marca da Ferramenta	Fornecedor ou Desenvolvida Internamente	Versão
Gerenciamento de Teste			
Controle de Defeitos			
Ferramenta ASQ para teste funcional			
Ferramenta ASQ para teste de desempenho			
Gerador de Perfil ou Monitor de Cobertura de Teste			
Gerenciamento de Projeto			
Ferramentas DBMS			

9.4 CONFIGURAÇÕES DO AMBIENTE DE TESTE

[Devem ser fornecidas e suportadas as seguintes Configurações de Ambiente de Teste para este projeto.]

Nome da Configuração	Descrição	Implementada na Configuração Física
Configuração do usuário comum		
Mínima configuração suportada		
Motivada por funções visuais e		

motoras		
Sistema Operacional Internacional de Dois Bytes		
Instalação de Rede (não cliente)		

10 MARCOS DA ITERAÇÃO

[Identifique os principais marcos da programação que definem o contexto do Esforço de Teste. Evite repetir muitos detalhes que já estejam documentados em outros lugares como, por exemplo, em planos que abordam o projeto inteiro.]

Marco	Data de início		Data de término	
	Planejada	Real	Planejada	Real
Plano de iteração acordado				
Início da iteração				
Elaboração da baseline dos requisitos				
Elaboração da baseline da arquitetura				
Elaboração da baseline da interface com o usuário				
Liberação do primeiro build para teste				
Aceitação do primeiro build para teste				
Termino do ciclo de teste do primeiro build				
Liberação do segundo build para teste				
Aceitação do segundo build para teste				
Termino do ciclo de teste do segundo build				
Liberação do terceiro build para teste				
Aceitação do terceiro build para teste				
Termino do ciclo de teste do terceiro build				
Liberação do enésimo build para teste				
Aceitação do enésimo build para teste				
Termino do ciclo de teste do enésimo build				

11 PROCEDIMENTOS E PROCESSOS DE GERENCIAMENTO

[Resuma os processos e os procedimentos que deverão ser usados quando surgirem problemas no Plano de Teste e em sua execução.]

11.1 MEDIÇÃO E AVALIAÇÃO DA EXTENSÃO DO TESTE

[Resuma o processo de medição e avaliação a ser usado para rastrear a extensão do teste.]

11.2 GERAÇÃO DE RELATÓRIOS SOBRE COBERTURA DE TESTE

[Resuma o processo de avaliação para revisar e aceitar os produtos liberados deste Plano de Teste.]

11.3 APROVAÇÃO E ENCERRAMENTO

[Resuma o processo de aprovação e liste os cargos (e os nomes dos ocupantes atuais) que deverão aprovar inicialmente o plano e encerre com a execução satisfatória do plano.]

APÊNDICE C.16 – PLANO DE IMPLANTAÇÃO

<Nome da Empresa>
<Slogan>

<Rua>, Nº <Número> - <Bairro> - <Cidade> - <UF>
CEP: <CEP>

<NOME DO PROJETO>

PLANO DE IMPLANTAÇÃO

VERSÃO 1

Histórico de revisão

Data	Versão	Descrição	Autor

SUMÁRIO

1 PLANO DE IMPLANTAÇÃO 307
1.1 INTRODUÇÃO 307
1.2 FINALIDADE 307
1.3 ESCOPO 307
1.4 DEFINIÇÕES, ACRÔNIMOS E ABREVIAÇÕES 307
1.5 VISÃO GERAL 307
1.6 REFERÊNCIAS 307
2 PLANEJAMENTO DE IMPLANTAÇÃO 308
2.1 RESPONSABILIDADES 308
2.2 PROGRAMAÇÃO 308
3 RECURSOS 309
3.1 INSTALAÇÕES 309
3.2 HARDWARE 309
4.3 UNIDADE DE IMPLANTAÇÃO 309
3.2.1 Software de suporte 309
3.2.2 Documentação de suporte 309
3.2.3 Pessoal de suporte 309
4 TREINAMENTO 310

1 PLANO DE IMPLANTAÇÃO

1.1 INTRODUÇÃO

[Forneça uma visão geral do documento inteiro.]

1.2 FINALIDADE

[Descreva a finalidade deste documento]

1.3 ESCOPO

[Identifique os destinatários dos itens identificados no Plano de Implantação.]

1.4 DEFINIÇÕES, ACRÔNIMOS E ABREVIAÇÕES

[Consultar o documento Glossário de Negócios.]

1.5 VISÃO GERAL

[Explique como este documento está organizado.]

1.6 REFERÊNCIAS

[Esta subseção fornece uma lista completa dos documentos. Identifique cada documento por título, número do relatório (se aplicável), e organização de publicação. Especifique as fontes a partir das quais as referências podem ser obtidas]

2 PLANEJAMENTO DE IMPLANTAÇÃO

[Descreva todas as atividades executadas na implantação do produto para o cliente. As atividades incluem planejamento, teste beta, preparação de itens a serem disponibilizados, empacotamento, "envio", instalação do produto, treinamento e suporte.]

2.1 RESPONSABILIDADES

[Identifique as responsabilidades do cliente e da equipe de desenvolvimento na preparação para a implantação. O mais importante nesta seção é a descrição do envolvimento do cliente nos testes de aceitação e no processo de tratamento de discrepâncias.]

2.2 PROGRAMAÇÃO

[Descreva o programa e os marcos para a condução das atividades de implantação. Os marcos de implantação devem ser compatíveis com os marcos do projeto.
Leve em consideração os seguintes detalhamentos do fluxo de trabalho de Implantação:

- *Planejamento da implantação*
- *Desenvolvimento do material de suporte*
- *Gerenciamento dos testes de aceitação*
  - *Testes de aceitação no local de desenvolvimento*
  - *Testes de aceitação no local de implantação*
- *Produção da unidade de implantação*
- *Gerenciamento do programa beta*
- *Gerenciamento da produção em massa e do empacotamento do produto*
- *Disponibilização do produto pela internet]*

3 RECURSOS

[Liste os recursos e suas fontes necessários para executar as atividades de implantação.]

3.1 INSTALAÇÕES

[Se aplicável, descreva as instalações necessárias para testar e implantar o software. As instalações podem incluir construções especiais ou salas com piso elevado, requisitos de energia elétrica e recursos especiais de suporte aos requisitos de privacidade e segurança.]

3.2 HARDWARE

[Identifique o hardware necessário para execução e suporte ao software, se aplicável. Especifique modelo, versões e configurações. Forneça informações sobre suporte do fabricante e licenças.]

4.3 UNIDADE DE IMPLANTAÇÃO

[Liste o software e a documentação fornecida como parte do produto liberado.]

3.2.1 Software de suporte

[Conforme aplicável, descreva todo o software necessário para suporte ao produto liberado, como ferramentas, compiladores, ferramentas de teste, dados de teste, utilitários, ferramentas de CM, bancos de dados, arquivos de dados e assim por diante.]

3.2.2 Documentação de suporte

[Conforme aplicável, descreva a documentação necessária para suporte ao produto liberado, como descrições de design, casos e procedimentos de teste, manuais do usuário etc.]

3.2.3 Pessoal de suporte

[Conforme aplicável, descreva o pessoal e os respectivos níveis de habilidades necessários para suporte ao produto liberado.]

4 TREINAMENTO

[Descreva o plano e as informações para treinamento dos usuários de forma que eles possam utilizar e adaptar o produto de acordo com suas necessidades. Caso possua um artefato Plano de Treinamento, referencie-o]

APÊNDICE C.17 – PLANO DE TREINAMENTO

<Nome da Empresa>
<Slogan>

<Rua>, Nº <Número> - <Bairro> - <Cidade> - <UF>
CEP: <CEP>

<NOME DO PROJETO>

PLANO DE TREINAMENTO

VERSÃO 1

Histórico de revisão

Data	Versão	Descrição	Autor

SUMÁRIO

1 PLANO DE TREINAMENTO 319
1.1 INTRODUÇÃO 319
1.2 PERÍODO 319
1.3 ESCOPO 319
1.4 OBJETIVO 319
1.5 GESTOR 319
1.6 CLIENTE 319
1.7 PRODUTO 319
2 DESAFIOS 320
3 IMPACTOS 321
4 FASES DO TREINAMENTO (OPCIONAL) 322
5 EMENTA 323
6 RELATÓRIO DO TREINAMENTO 325
7 CONCLUSÃO 326

1 PLANO DE TREINAMENTO

1.1 INTRODUÇÃO

[Forneça uma visão geral do documento inteiro.]

1.2 PERÍODO

[Colocar o período de treinamento, com data de previsão do termino.]

1.3 ESCOPO

[Breve descrição da utilidade do documento, do que é afetado e/ou influenciado por ele.]

1.4 OBJETIVO

[Descrição do objetivo a ser alcançado.]

1.5 GESTOR

[Responsável pelo documento de Plano de Treinamento e pelo treinamento]

1.6 CLIENTE

[Empresa, pessoa ou instituição que irá receber o treinamento].

1.7 PRODUTO

[Produto que será passado para o cliente durante o treinamento].

2 DESAFIOS

[Desafios, em vista, que poderão ocorrer ao longo do treinamento].

3 IMPACTOS

[Impactos que ocorrer durante o treinamento.]

4 FASES DO TREINAMENTO (OPCIONAL)

[Caso exista a necessidade de dividir o treinamento em fases, especificar aqui como se dará essas fases.]

5 EMENTA

[Anexar a esse documento a ementa usada no treinamento. Com a especificação do conteúdo, hora aula, recursos usados, dia e etc. Pode ser usada como padrão uma ementa elaborada pela UFRJ, mostrada abaixo.]

EMENTA DE CURSO

IDENTIFICAÇÃO DO CURSO

Nome: *[Nome da disciplina]*

Código: *[Código da disciplina]*

Professor(es): *[Professor 1; Professor 2; Professor n]*

Professor(es) colaborador(es): *[Professor 1; Professor 2; Professor n]*

Carga horária: *[XX]* hora(s)

Número de créditos: *[XX (quando necessário)]*

Nível: *[Fundamental | Médio | Técnico | Superior | Especialização]*

Pré-requisitos: *[Inexiste | Existente: Descreva o(s) pré-requisito(s)]*

Período de atividades: *[Diário | Semanal | Quinzenal | Mensal | Trimestral | Semestral]*

Endereço da disciplina na WEB: *[endereço eletrônico do site da disciplina]*

OBJETIVOS

[Descrever textualmente os objetivos gerais e específicos da disciplina]

EMENTA

[Faça um breve resumo da ementa, de forma textual]

PROGRAMA

- *[Primeiro assunto ou tema a ser trabalhado]*
- *[Segundo assunto ou tema a ser trabalhado]*
- *[Terceiro assunto ou tema a ser trabalhado]*
- *[Enésimo assunto ou tema a ser trabalhado]*

METODOLOGIA DE ENSINO

[Descreva de que forma serão ministradas as aulas, quais as técnicas empregadas e os recursos]

SISTEMA DE AVALIAÇÃO

[Descreva detalhadamente como será avaliado o conhecimento dos alunos]

REFERÊNCIAS BIBLIOGRÁFICAS

[Liste aqui as referências bibliográficas, no padrão ABNT, usadas como referencial teórico e prático para a disciplina]

6 RELATÓRIO DO TREINAMENTO

[Um relatório de treinamento especifica os resultados obtidos de um treinamento realizado. Através os gestores da empresa e/ou o cliente podem avaliar se os resultados foram satisfatórios. Portanto, abaixo está expresso um modelo desse relatório. É recomendável criá-lo fora deste documento e anexá-lo ao final do treinamento proposto por este plano.]

RELATÓRIO DE TREINAMENTO

Este documento tem como intuito relatar as atividades realizadas durante o treinamento ocorrido no período de *[data inicial <dd/mm/aaaa>]* à *[data final <dd/mm/aaaa>]*, na cidade de *[Nome da cidade - UF]*, para os funcionários e contratados da *[nome da entidade que contratou o treinamento]*, listados abaixo. Foram feitas atividades praticas e teóricas sobre *[liste todas as atividades realizadas durante o treinamento]*. Em seguida são relacionadas as atividades realizadas no decorrer do curso.

Atividade	Aproveitamento
[Primeira atividade]	*[Bom | Aceitável | Insuficiente]*
[Segunda atividade]	
[Terceira atividade]	
[Enésima atividade]	

Parecer do relator:_____

_____

_____

_____

<Cidades>, 12 de March de 2010

_____ _____

Assinatura do cliente Assinatura do Analista de Suporte

7 CONCLUSÃO

[Após o treinamento deverá ser escrita a conclusão, onde deverá ser colocado tudo que for relevante, os de forma a melhorar próximos treinamentos.]

APÊNDICE C.18 – GESTÃO DE AÇÕES CORRETIVAS

<Nome do projeto>

Gestão de ações corretivas

[Nome do gerente do projeto]

<Nome da Empresa>

<Slogan>

<Rua>, Nº <Número> - <Bairro> - <Cidade> - <UF>

Fase	Desvio	Descrição do desvio	Prioridade	Responsável
[Fase da ocorrência]	*[Onde ocorreu o desvio]*	*[Descrição do desvio]*	*[Alta | Média | Baixa]*	*[Nome do responsável pela ocorrência do desvio]*

Impacto	Data da ocorrência	Status	Ação corretiva	Responsável pela ação	Prazo	Observações
[Impacto causado pelo desvio]	*[dd/mm/aaaa]*	*[Corrigido | Em aberto | Escalado para <cargo superior>]*	*[Descrição de qual ação tomada quando o desvio ocorrer]*	*[Responsável pela ação]*	*[dd/mm/aaaa]*	

APÊNDICE C.19 – LISTA DE VERIFICAÇÃO DA QUALIDA

<Nome do projeto>

Lista de verificação de qualidade

<Nome da Empresa>
<Slogan>

<Rua>, Nº <Número> - <Bairro> - <Cidade>

Data da verificação: *[dd/mm/aaaa]*

Informações

Gerente de Projetos:	*[Nome do gerente do projeto]*
Auditor:	*[Nome do auditor ou analista de qualidade]*
Fase verificada:	*[Nome da fase verificada]*
Auditor:	*[Nome do auditor ou analista de qualidade]*

Fases do ciclo de vida	Artefato	Descrição da verificação	Resultado	Responsável	Prazo para correção	Observações
Primeira Fase	*[Sigla 1]*	*[Primeira verificação realizada]*	*[Conforme | Não-conforme | Não se Aplica | Pendente]*	*[Nome do responsável pela correção]*	*[dd/mm/aaaa]*	
	[Sigla 2]	*[Segunda verificação realizada]*				
	[Sigla 3]	*[Terceira verificação realizada]*				
	[Sigla n]	*[Enésima verificação realizada]*				
Enésima Fase						